Adobe

# Adobe LiveCycle Production Print ES2

**Version 9.0**

## Benutzerhandbuch

**Rev B**

Powered by StreamServe™

Adobe LiveCycle Production Print ES2 Benutzerhandbuch
Rev B

© 2001-2009 STREAMSERVE, INC.
ALLE RECHTE VORBEHALTEN
US-Patentnummer 7.127.520

Ohne ausdrückliche schriftliche Genehmigung von StreamServe, Inc. darf kein Teil dieses Dokuments in einer beliebigen Form oder mit beliebigen Hilfsmitteln zu einem beliebigen Zweck auf elektronischem oder mechanischem Wege vervielfältigt oder übertragen werden. Bei den in diesem Dokument enthaltenen Informationen sind Änderungen vorbehalten. StreamServe, Inc. übernimmt keine Haftung für Folgeschäden aus Fehlern oder Ungenauigkeiten, die in diesem Handbuch auftreten können. Alle eingetragenen Namen, Produktnamen und Marken von anderen Unternehmen werden in dieser Dokumentation lediglich zur Identifizierung verwendet und sind Eigentum des jeweiligen Unternehmens. Unternehmen, Namen und Daten, die in Beispielen in diesem Dokument verwendet werden, sind, sofern nicht anders angegeben, erfunden.

StreamServe, Inc. übernimmt keine Garantie, Verantwortung oder Haftung jeglicher Art bezüglich Produkten und Diensten von Drittanbietern einschließlich der Haftung für Schäden, die aufgrund der Inkompatibilität von Drittanbieterprodukten und -diensten mit den Produkten und Diensten von StreamServe, Inc. auftreten. Durch die Verwendung von StreamServe-Produkten und in diesem Dokument aufgeführten Drittanbieterprodukten erklären Sie sich damit einverstanden, StreamServe, Inc. bezüglich Drittanbieterprodukten und -diensten nicht für Schäden verantwortlich oder haftbar zu machen und auch nicht die Absicht zu haben, dies zu tun.

Die Marken, Logos und Dienstleistungsmarken in diesem Dokument sind Eigentum von StreamServe, Inc. oder von Drittanbietern. Es ist unzulässig, die Marken ohne ausdrückliche schriftliche Genehmigung durch StreamServe, Inc. oder den als Eigentümer der Marken auftretenden Drittanbieter zu verwenden.

Die Verwendung des StreamServe-Produkts mit Drittanbieterprodukten, die in diesem Dokument nicht erwähnt werden, erfolgt auf eigenes Risiko – auch in Bezug auf die StreamServe-Produkte.

StreamServe-Website
http://www.streamserve.com

Adobe® LiveCycle® Production Print ES2

Die Lizenz für dieses Produkt wurde von Adobe Systems Incorporated oder von einem von Adobe autorisierten Dritten erworben.

Es handelt sich um ein lizenziertes Produkt, das Technologie von StreamServe Inc. enthält.

Die Nutzung dieser Software ist durch die Endbenutzer-Lizenzvereinbarung (EULA) von Adobe Systems Incorporated geregelt.

Sämtliche Wartungs- und Supportleistungen werden von Adobe Systems Incorporated oder einem von Adobe autorisierten Fremdanbieter bereitgestellt.

Die Bedingungen, die Ihre Nutzung der Software regeln, sind in der Endbenutzer-Lizenzvereinbarung beschrieben, die Sie zusammen mit dem von Adobe Systems Incorporated bereitgestellten Produkt erhalten.

Bei Fragen zur Lizenzierung wenden Sie sich bitte an Adobe Systems Incorporated.

Das Installationsmedium und das Dokumentationsset enthalten und verweisen auf Komponenten, die möglicherweise nicht für die Verwendung im Rahmen Ihrer Lizenz für Adobe LiveCycle Production Print ES2 aktiviert sind.

Adobe Systems Incorporated – Kontaktinformationen

Patch-Aktualisierungen, technische Hinweise und zusätzliche Informationen zu diesem Produkt können Sie von Ihrem Adobe Enterprise Support-Anbieter beziehen oder von der folgenden Website herunterladen:

www.adobe.com/support/products/enterprise/index.html

Antworten auf andere allgemeine Fragen erhalten Sie unter

http://www.adobe.com/aboutadobe/contact.html

Informationen für Entwickler

Auf der Adobe LiveCycle Developer Center-Website (www.adobe.com/devnet/livecycle) erhalten Sie aktuelle Informationen für Entwickler und haben die Möglichkeit, Ihre Kenntnisse mithilfe von Artikeln, angeleiteten Übungen, Codebeispielen, Downloads und Musteranwendungen zu erweitern.

Informationen zu verfügbaren Ressourcen für Entwickler finden Sie unter www.adobe.com/enterprise/developer/main.html.

Adobe, das Adobe-Logo und LiveCycle sind Marken oder eingetragene Marken von Adobe Systems Incorporated in den USA und/oder anderen Ländern. Alle anderen Marken sind Eigentum der jeweiligen Inhaber.

© 2007 - 2009 Adobe Systems Incorporated. Alle Rechte vorbehalten.

# Inhalt

**Informationen zu LiveCycle Production Print ES2 ..... 5**

**Einführung ..... 7**
**Vorlagen ..... 8**
**Externe Referenzen auf Fragmente und Bilder ..... 9**
**Nachbearbeitung ..... 10**
Ausführen von Skripts vor und nach jedem Datensatz ..... 10
Logische Dokumentgruppen – Document-Trigger ..... 10
**Überlegungen zur Leistung ..... 12**
Zwischenspeicherung der Schriftartdaten ..... 12
Vorlagenzwischenspeicherung ..... 12
**Unterstützte Ausgabeformate ..... 13**
**Silbentrennung ..... 14**

**Verwenden einer vorhandenen Formularvorlage ..... 15**
**Konfigurieren des Projekts ..... 16**
**Bereitstellen und Ausführen ..... 18**

**Verwenden einer neu erstellten Formularvorlage ..... 19**
**Konfigurieren des Projekts ..... 20**
**Bereitstellen und Ausführen des Projekts ..... 22**

**Verwenden von Vorlagen ..... 23**
**Verwenden einer StreamServe-Variablen zum Laden der Vorlage ..... 24**
**Verwenden eines SOM-Ausdrucks zum Laden einer Vorlage ..... 26**
**Laden einer Vorlage direkt aus dem LiveCycle-Repository ..... 27**
**Laden von Schriftarten ..... 28**

**Arbeiten mit externen Vorlagen ..... 29**

**Integration mit LiveCycle Production Print ES2 und LiveCycle ES2 ..... 31**
**Zugriff auf das Adobe LiveCycle ES2-Repository ..... 32**
Herstellen von Verbindungen zu LiveCycle-Repositorys ..... 32
Importieren einer Ressource aus dem Adobe LiveCycle ES2-Repository ..... 34
Aktualisieren von Ressourcen aus dem Adobe ES2 LiveCycle-Repository ..... 35
**Aufrufen von LiveCycle Production Print ES2 aus LiveCycle ES2 ..... 37**
Merkmale der LiveCycle Production Print DSC ..... 38
Post ..... 38
Run ..... 39
Generate ..... 40
Fehlerbehandlung ..... 41
StreamServer-Konfiguration ..... 42
Erstellen eines Dienstanforderungs-Eingabe-Connectors ..... 43
Aktivieren von Dienstantworten ..... 43
Abrufen von Variablen ..... 44

**Aufrufen von LiveCycle ES2-Prozessen aus LiveCycle Production Print ES2 ..... 45**
LiveCycle-Ausgabe-Connector ..... 45
Erstellen eines Ausgabe-Connector-fähigen LiveCycle ES2-Process ..... 46
Erstellen eines LiveCycle-Ausgabe-Connectors ..... 47
Verwendungsszenario ..... 50
LiveCycle-Filter ..... 51
Erstellen eines filterfähigen LiveCycle ES2-Process ..... 52
Erstellen eines LiveCycle-Filters ..... 53
Verwendungsszenario ..... 56
Globale LiveCycle-Filter- und LiveCycle-Connector-Einstellungen ..... 57

**Das Musterprojekt ..... 59**
**Ausführen des Musterprojekts ..... 60**
**Überprüfen der Basisinstallation ..... 61**
**Musterprojektkonfigurationen ..... 62**
Die Platform-Konfiguration ..... 63
Das Ressourcenset ..... 65
Die Messages ..... 65
Die Runtime-Konfigurationen ..... 66

**Unterstützung für Etikettendrucker ..... 69**
**ZPL II ..... 70**
Textausgabe – ZPL II ..... 70
Barcodeunterstützung – ZPL II ..... 70
RFID-Unterstützung – ZPL II ..... 71
Unterstützung von Benutzeroberflächenobjekten – ZPL II ..... 71
**Intermec FP/DP ..... 72**
Textausgabe – Intermec FP/DP ..... 72
Barcodeunterstützung – Intermec FP/DP ..... 73
RFID-Unterstützung – Intermec FP/DP ..... 73
Unterstützung für Benutzeroberflächenobjekte – Intermec FP/DP ..... 74
**Intermec IPL ..... 75**
Textausgabe – Intermec IPL ..... 75
Barcodeunterstützung – Intermec IPL ..... 76
RFID-Unterstützung – Intermec IPL ..... 76
Unterstützung von Benutzeroberflächenobjekten – Intermec IPL ..... 77
**Printronix PGL/IGP ..... 78**
Textausgabe – Printronix PGL/IGP ..... 78
Barcodeunterstützung – Printronix PGL/IGP ..... 78
RFID-Unterstützung – Printronix PGL/IGP ..... 79
Unterstützung für Benutzeroberflächenobjekte – Printronix PGL/IGP ..... 79
**TEC ..... 80**
Textausgabe – TEC ..... 80
Barcodeunterstützung – TEC ..... 81
RFID-Unterstützung – TEC ..... 81
Unterstützung für Benutzeroberflächenobjekte – TEC ..... 81

**Referenz für die Benutzeroberfläche ..... 83**
**LiveCycle Designer ES2 Process-Tool ..... 84**
Befehle im Menü „Datei“ ..... 85
Dialogfeld „Einstellungen“ ..... 86
Dialogfeld „Vorlage auswählen“ ..... 89
**Design Center ..... 91**
Befehle im Menü „Extras“ ..... 91
Befehle im Menü „Ressourcen“ ..... 91
Dialogfelder ..... 92
Dialogfeld „Aktive Adobe LiveCycle-Repositoryverbindung auswählen“ ..... 92
Dialogfeld „Adobe LiveCycle-Repositoryressource importieren“ ..... 93
Dialogfeld „Alle Adobe LiveCycle-Repositoryressourcen aktualisieren“ ..... 95
Dialogfeld „Runtime-Process-Einstellungen“ – Registerkarte „Allgemein“ ..... 98
Beim Zugriff auf das LiveCycle ES2-Repository verwendete Symbole ..... 99

**Skripts ..... 101**
**Übernehmen eines SOM-Ausdruckswerts in eine StreamServe-Variable ..... 102**
Informationen zu SOM-Ausdrücken ..... 104
**Zugriff auf StreamServe-Variablen im XFA-Prozessor ..... 105**
**Verwenden von Vor- und Nach-Process-Skripts ..... 106**
**Schreiben in die StreamServe-Protokolldatei ..... 108**

**XFA-Unterstützung ..... 109**
**XFA-Elemente ..... 110**
Vollständig unterstützte XFA-Elemente ..... 110
XFA-Elemente mit nicht unterstützten Attributen ..... 111
XFA-Elemente für das Hinzufügen von Daten zu Tagged-PDF-Dateien ..... 113
Nicht unterstützte XFA-Elemente ..... 114
**XFA-Skripts ..... 115**
Unterstützte Skriptfunktionen ..... 115
Unterstützte Skripteigenschaften ..... 118
Unterstützte Skriptobjektmodelle ..... 125
Unterstützte Events ..... 125
Überlegungen zur Verwendung von Skripts ..... 125

**Glossar ..... 127**

# Informationen zu LiveCycle Production Print ES2

Mit Adobe® LiveCycle® Production Print ES2 können Unternehmen die Anforderungen an die Datencenterausgabe auf Produktionsebene erfüllen und unter Verwendung von XML-Daten aus Kernanwendungen dynamisch personalisierte Dokumente für die Ausgabe in verschiedenen Formaten (z. B. AFP, PostScript®, PDF oder PCL) generieren. LiveCycle Production Print ES2 erweitert den Funktionsumfang von LiveCycle® Designer ES2 sowie der LiveCycle ES2-Plattform insgesamt und stellt eine einheitliche Designumgebung für die gesamte geschäftskritische Kommunikation bereit.

LiveCycle Production Print ES2 implementiert leistungsfähige Druckfunktionen für die Adobe LiveCycle ES2-Produktgruppe. Es handelt sich hierbei um eine integrierte Lösung von StreamServe und Adobe, bei der Adobe LiveCycle Designer ES2 in die StreamServe-Umgebung integriert wird.

Dieses Dokument enthält spezifische Informationen zu LiveCycle Production Print ES2. Allgemeine Informationen zu StreamServe finden Sie in der StreamServe-Standarddokumentation. Der Leser sollte Grundkenntnisse über StreamServe besitzen und idealerweise am Kurs „StreamServe Persuasion Fundamentals“ teilgenommen haben.

### StreamServe Persuasion

LiveCycle Production Print ES2 basiert auf StreamServe Persuasion4, der Enterprise Document Presentment-Software (EDP) von StreamServe. StreamServe Persuasion umfasst folgende Komponenten:

- Das StreamServe Component Framework mit der Basissoftware zum Ausführen von StreamServe-Anwendungen.
- StreamServer, die Software zum Ausführen von StreamServer-Anwendungen.
- Design Center, das zentrale Design-Tool von StreamServe Persuasion. In Design Center können Sie folgende Schritte ausführen:
  - Erstellen Ihrer StreamServe-Projekte
  - Konfigurieren der Verbindung zur Quellanwendung
  - Identifizieren und Extrahieren der Eingabedaten
  - Konfigurieren der Ausgabe an die Ausgabegeräte

### Benutzerdokumentation

Neben der LiveCycle Production Print ES2-spezifischen Dokumentation ist natürlich auch die StreamServe-Standarddokumentation relevant.

Die folgenden Dokumente beziehen sich speziell auf LiveCycle Production Print ES2:

- Versionshinweise: Beschreibt die neuen und veränderten Funktionen.
- Benutzerhandbuch (dieses Dokument): Enthält ausführliche Informationen zu den LiveCycle Production Print ES2-Funktionen.
- Installationshandbuch: Enthält Hinweise zum Installieren, Aktualisieren und Überprüfen der LiveCycle Production Print ES2-Software. Darüber hinaus werden unterstützte Softwareprogramme und Plattformen aufgelistet.

### Musterprojekt

Im Lieferumfang der StreamServe-Installation ist ein Musterprojekt enthalten (siehe *Das Musterprojekt* auf Seite 59).

# Einführung

Adobe® LiveCycle® Designer ES2 ist das Adobe-Tool für den Formularentwurf, das in die StreamServe-Designumgebung integriert und als StreamServe-Process-Tool genutzt wird. Die Ausgabe wird in verschiedenen Formaten erstellt, die optisch mit der entsprechenden Ausgabe aus LiveCycle Output ES2 identisch sind.

Der Zugriff von StreamServe Design Center auf Adobe LiveCycle Designer ES2 erfolgt entweder über eine eigenständige Installation oder über eine Paketversion mit LiveCycle Workbench ES2.

Formularvorlagen können mit einem XML-Instanzdokument zusammengeführt werden. In StreamServe Design Center können Sie auch eine Formularvorlage (mit Bindungen zu anderen Datenformaten als XML) ganz neu anlegen.

Das Process-Tool für LiveCycle Designer ES2 kann für folgende Zwecke verwendet werden:

- Importieren einer LiveCycle Designer ES2-Formularvorlage.
- Erstellen einer neuen LiveCycle Designer ES2-Formularvorlage.
- Bearbeiten einer Formularvorlage mithilfe von LiveCycle Designer ES2.
- Exportieren von Vorschaudateien für andere Datenformate als XML zur Verwendung mit einer eigenständigen LiveCycle Designer ES2-Instanz.

### Ausführen ohne Adobe LiveCycle Designer ES2

Wenn Sie eine bestehende Formularvorlage verwenden, können Sie das Process-Tool ausführen, ohne dass LiveCycle Designer ES2 installiert ist. In diesem Fall ist es jedoch nicht möglich, die Formularvorlage anschließend in der StreamServe-Designumgebung zu bearbeiten.

### In diesem Kapitel behandelte Themen


- *Vorlagen* auf Seite 8
- *Externe Referenzen auf Fragmente und Bilder* auf Seite 9
- *Nachbearbeitung* auf Seite 10
- *Überlegungen zur Leistung* auf Seite 12
- *Unterstützte Ausgabeformate* auf Seite 13
- *Silbentrennung* auf Seite 14

# Vorlagen

Vorlagen können wahlweise aus einem Ressourcenset im Design Center-Projekt oder aus dem LiveCycle ES2-Repository geladen werden.

### Aus Design Center-Ressourcenset

Vorlagen, die aus einem Ressourcenset im Design Center-Projekt geladen werden, sind statisch mit dem Process verknüpft. Der Ladevorgang erfolgt beim Start.

### Aus LiveCycle-Repository

Zur Behandlung von aus dem LiveCycle-Repository geladenen Vorlagen stehen verschiedene Möglichkeiten zur Auswahl:

- Sie können die Vorlagen im Design Center-Ressourcenset speichern und zur Entwurfszeit laden.
  Design Center stellt eine Verbindung zum LiveCycle-Repository her und lädt die Vorlagen in das Design Center-Ressourcenset. Aktualisierte Vorlagen werden wieder aus dem LiveCycle-Repository geladen.
- Sie können mithilfe von StreamServe-Variablen oder SOM-Ausdrücken zur Laufzeit eine Verbindung zum LiveCycle-Repository herstellen und die Vorlagen dynamisch laden. Die Vorlagen können einmal pro Job oder einmal für jeden verarbeiteten Datensatz geladen werden.

Informationen zum dynamischen Abrufen und Laden von Vorlagen finden Sie unter *Verwenden von Vorlagen* auf Seite 23.

### Überlegungen zur Leistung

Beachten Sie, dass die Systemleistung durch das dynamische Laden von Vorlagen erheblich beeinträchtigt werden kann. Die Leistungsparameter hängen stark vom Dateisystem, der Datei-E/A und der Netzwerkkapazität ab. Um eine optimale Systemleistung zu gewährleisten, müssen Sie einen schnellstmöglichen Zugriff des Process auf die Vorlagen sicherstellen.

Zur Verbesserung der Leistung bei der Verwendung dynamischer Vorlagen können Sie die Vorlagenzwischenspeicherung aktivieren. Informationen hierzu finden Sie unter *Vorlagenzwischenspeicherung* auf Seite 12.

# Externe Referenzen auf Fragmente und Bilder

Es werden XDP-Dateien mit Abhängigkeiten (externe Verweise auf Fragmente und Bilder) unterstützt. Referenzierte Dateien sind nicht in die XDP-Hauptdatei eingebettet. Wenn Sie eine Vorlage mit Abhängigkeiten importieren, löst das Process-Tool die Verweise auf und importiert die erforderlichen Ressourcen.

StreamServer kann Abhängigkeiten aus dem lokalen Dateisystem, aus dem lokalen Netzwerk, aus HTTP-Ressourcen und aus FTP-Ressourcen verarbeiten.

Wenn Sie eine LiveCycle Archive(LCA)-Datei importieren, sind das Haupt-XDP und alle zugehörigen Abhängigkeiten im importierten LCA-Paket enthalten.

Die Vorlagen und dazugehörigen Dateien können aus dem Adobe LiveCycle-Repository in ein Ressourcenset im Design Center-Projekt importiert werden. Informationen hierzu finden Sie unter *Zugriff auf das Adobe LiveCycle ES2-Repository* auf Seite 32.

**Hinweis:** Beim Aktualisieren der importierten Ressourcen an deren Quellspeicherort werden die in das Process-Tool importierten Abhängigkeiten nicht automatisch aktualisiert.
Wenn Abhängigkeiten am Quellspeicherort geändert werden, müssen Sie diese im Process-Tool aktualisieren.

# Nachbearbeitung

Bei der Nachbearbeitung in LiveCycle Production Print ES2 wird die Ausgabe auf dieselbe Weise erzeugt wie bei Verwendung des StreamServe PageOUT-Tools.

Bei der Verwendung einer Formularvorlage mit einem XML-Datensatz mit mehreren Datensätzen wird bei der Nachbereitung nicht jeder Datensatz als separate Message behandelt. Stattdessen werden Datensätze als Batch innerhalb eines Process behandelt.

## Ausführen von Skripts vor und nach jedem Datensatz

Sie können Vor- und Nach-Process-Skripts vor und nach jedem Datensatz ausführen. Dazu müssen Sie die entsprechende Option im Dialogfeld **Einstellungen** aktivieren. Informationen hierzu finden Sie unter *Dialogfeld „Einstellungen“* auf Seite 86.

Das erste Vor-Process-Skript und das letzte Nach-Process-Skript werden in der Vor- bzw. Nach-Process-Phase ausgeführt. Alle anderen Vor- und Nach-Process-Skripts werden ausschließlich in der Process-Phase ausgeführt. Dies ist beim Konfigurieren der Skripts zu beachten.

Informationen hierzu finden Sie unter *Verwenden von Vor- und Nach-Process-Skripts* auf Seite 106.

## Logische Dokumentgruppen – Document-Trigger

Normalerweise wird jeder Datensatz in LiveCycle Production Print ES2 automatisch einem Dokument zugeordnet.

Mithilfe eines Document-Triggers können Sie festlegen, wie die Ausgabe der mit dem Ausgabe-Connector verbundenen Processes geteilt und gruppiert werden soll. Dies wäre beispielsweise sinnvoll, wenn alle Dokumente mit derselben Kundennummer im Eingabe-Job demselben Document zugeordnet werden sollen. Der Document-Trigger ist in den Runtime-Ausgabe-Connector-Einstellungen definiert. Weitere Informationen finden Sie in der *StreamServer Design Center-Dokumentation*.

Document-Triggers können für die Ausgabemodi „Document“ und „Job“ verwendet werden.

### Verwenden des automatischen Document-Triggers

Mithilfe der Einstellung **Automatischer Doc-Trigger** können Sie festlegen, dass jeder Datensatz automatisch einem Dokument zugeordnet wird. Damit wird die Document-Triggervariable deaktiviert.

Bei aktualisierten Projekten aus Versionen vor LiveCycle Production Print ES2 ist diese Einstellung standardmäßig ausgewählt.

Informationen hierzu finden Sie unter *Dialogfeld „Runtime-Process-Einstellungen“ – Registerkarte „Allgemein“* auf Seite 98.

### So legen Sie den automatischen Document-Trigger fest

1. Klicken Sie in der Runtime-Ansicht mit der rechten Maustaste auf den LiveCycle Designer ES2-Process, und wählen Sie **Einstellungen**.
   Das Dialogfeld **Runtime-Process-Einstellungen** wird geöffnet.
2. Wählen Sie die Registerkarte **Allgemein**.
3. Wählen Sie die Option **Automatischer Doc-Trigger**.
   Die **Document-Triggervariable** (angegeben auf der Registerkarte **Document-Trigger** des Dialogfelds **Runtime-Connector-Einstellungen**) ist damit deaktiviert, sodass jeder Datensatz automatisch einem Dokument zugeordnet wird. Eine Gruppierung der Ausgabe in logische Dokumente erfolgt in diesem Fall nicht.

# Überlegungen zur Leistung

## Zwischenspeicherung der Schriftartdaten

Zur Verbesserung der Leistung können Sie das Programmstart-Argument `preloadmorefontdata` verwenden, das die Zwischenspeicherung zusätzlicher Schriftartdaten beim Programmstart aktiviert. Dies führt zu einer schnelleren Laufzeitausführung, jedoch auch zu einer längeren Startzeit und einem erhöhten Speicherverbrauch.

Standardmäßig werden keine zusätzlichen Schriftartdaten zwischengespeichert.

Weitere Informationen finden Sie in der Dokumentation *StreamServe Startup arguments*.

## Vorlagenzwischenspeicherung

Das Laden und Entladen von Formularvorlagen kann sich auf die Leistung auswirken. Wenn die gleiche Vorlage mehrmals verwendet wird, wie es bei der Verwendung von dynamischen Vorlagen der Fall ist, kann die Leistung durch Zwischenspeicherung erheblich verbessert werden.

Das Zwischenspeichern von Vorlagen wird im Dialogfeld **Einstellungen** aktiviert bzw. deaktiviert. Informationen hierzu finden Sie unter *Dialogfeld „Einstellungen“* auf Seite 86.

Bei Änderungen an der ursprünglichen Formularvorlage wird diese automatisch erneut aus dem Cache geladen. Anhand des Zeitstempels der ursprünglichen Vorlagendatei und der zwischengespeicherten Datei wird ermittelt, ob die ursprüngliche Datei geändert wurde. Dies gilt nur für die Hauptvorlagendatei. Geänderte Fragmente werden nicht erneut geladen.

Die Größe des Vorlagencaches kann im Control Center und über die Befehlszeile festgelegt werden. Die Cachegröße wird in KB angegeben, sie überschreibt den Standardwert für die Cachegröße.

Die Standardgröße des Caches beträgt 10240 KB.

### Control Center

Die Option **Cachegröße** wird in der Eigenschaftenansicht festgelegt.

### Befehlszeile

`-maxcachesize <Wert>`

# Unterstützte Ausgabeformate

Der XFA-Prozessor unterstützt die folgenden Ausgabeformate und Treiber.

| **Ausgabeformat** | **LiveCycle Production Print-Treiber** |
|---|---|
| AFP – Advanced Function Presentation | AFP |
| IJPDS – Ink Jet Printer Data Stream | IJPDS |
| PDF – Portable Document Format | PDF |
| Tagged PDF | PDF |
| PDF/A 1a | PDF |
| PDF/A 1b | PDF |
| PCL – Printer Control Language | PCL5 |
| P.S. – Postscript | PostScript |
| TIFF – Tagged Image File Format | TIFF |
| ZPL II – Zebra Programming Language | Zebra ZPLII |
| FP/DP – Intermec Fingerprint/ Direct Protocol | Intermec FP/DP |
| IPL – Intermec Printer Language | Intermec IPL |
| PGL/IGP – Intelligent Graphics Printing/ Printronix Graphics Language | Printronix PGL/IGP |
| Toshiba TEC | TEC |
| Windows Print API | Windows-Treiber (StreamServe 3.x) |

Andere StreamServe Persuasion-Ausgabeformate werden nicht getestet, möglicherweise funktionieren sie jedoch mit Einschränkungen bezüglich der Schriftart- und Objektdarstellung.

Informationen zu den von StreamServe Persuasion SP4 unterstützten Ausgabeformaten finden Sie in der StreamServe-Standarddokumentation.

### Windows-Treiber

Zur Verwendung mit bestimmten Druckertreibern von Drittanbietern können Sie eine Adobe LiveCycle Production Print ES2-Treiberkonfigurationsdatei (`.drs`) für den Adobe LiveCycle Production Print ES2-Windows-Treiber erstellen. Informationen dazu finden Sie in der StreamServe-Dokumentation zu *Gerätetreiber-Tools*.

# Silbentrennung

LiveCycle Production Print ES unterstützt die Silbentrennung etwa wie Output ES2 und Forms ES2.

Informationen zu bekannten Einschränkungen im Hinblick auf die Silbentrennung finden Sie in den *Adobe LiveCycle Production Print ES2-Versionshinweisen*.

# Verwenden einer vorhandenen Formularvorlage

In diesem Kapitel wird beschrieben, wie Sie eine vorhandene Formularvorlage für den Ausdruck in der Produktionsumgebung aktivieren. Das XML-Instanzdokument wird mit einer Formularvorlage zusammengeführt.

### Funktionsweise in StreamServer

1. Das XMLIN-Event erkennt das XML-Instanzdokument und löst den Process aus.
2. Der Process führt den Inhalt des Eingabe-XML-Instanzdokuments mit der Formularvorlage zusammen und erzeugt strukturierte Ausgabedaten.
3. Die erzeugten Ausgabedaten werden zur Formatierung an den Treiber und die Nachbearbeitung gesendet.
4. Die formatierte Ausgabe wird an das betreffende Ausgabeziel gesendet.

### In diesem Kapitel behandelte Themen


- *Konfigurieren des Projekts* auf Seite 16.
- *Bereitstellen und Ausführen* auf Seite 18.

# Konfigurieren des Projekts

Voraussetzungen:

- Ein XML-Instanzdokument als Eingabedaten.
- Eine Formularvorlage mit Datenbindungen zum XML-Instanzdokument.
- Ein in Design Center geöffnetes StreamServe-Projekt.

### So fügen Sie die Formularvorlage als Ressource hinzu

1. Doppelklicken Sie im Projekt-Browser von Design Center auf das Ressourcenset. Die Ressourcenset-Ansicht wird geöffnet.
2. Klicken Sie mit der rechten Maustaste auf das Ressourcenset, und wählen Sie die Option **Importieren** und anschließend die als Ressource zu importierende Datei aus. Die Ressource wird erstellt und dem Ressourcenset hinzugefügt.

### So erstellen Sie die Message

1. Wählen Sie **Datei > Neu > Message**. Es wird eine leere Message erstellt.
2. Benennen Sie die Message um.

### So erstellen Sie das Event

1. Klicken Sie mit der rechten Maustaste auf eine beliebige Stelle in der Message-Ansicht, und wählen Sie **Event hinzufügen > XMLIN**. Der Message-Ansicht wird ein neues Event hinzugefügt.
2. Benennen Sie das Event um.
3. Klicken Sie mit der rechten Maustaste auf das Event, und wählen Sie die Option **Öffnen** aus. Das Event-Tool wird geöffnet.
4. Öffnen Sie das XML-Instanzdokument als Muster, indem Sie die Option **Datei > Muster öffnen** wählen. Das Dialogfeld **Ressource auswählen** wird geöffnet.
5. Wählen Sie die Datei aus. Die Datei wird in der XMLIN-Musteransicht angezeigt.
6. Erstellen Sie mit dem Muster-Tool ein Muster zum Auslösen des Events. Weitere Informationen finden Sie in der Dokumentation zu *StreamServe XMLIN*.
7. Speichern Sie Ihre Eingaben, und beenden Sie das Event-Tool.

### So erstellen Sie den Process

1. Klicken Sie mit der rechten Maustaste auf eine beliebige Stelle in der Message-Ansicht, und wählen Sie die Option **Process hinzufügen > Adobe LiveCycle Designer ES2** aus. Der Message-Ansicht wird ein neuer Process hinzugefügt.
2. Benennen Sie den Process um.

3 Klicken Sie mit der rechten Maustaste auf den Process, und wählen Sie die Option **Öffnen** aus. Das Process-Tool wird geöffnet und das Dialogfeld **Einstellungen** angezeigt. Informationen hierzu finden Sie unter *Dialogfeld „Einstellungen“* auf Seite 86.

4 Klicken Sie auf **LiveCycle Designer-Oberfläche beim Start laden** und dann auf **OK**. Hinweis: Dieser Schritt ist optional, da LiveCycle Designer möglicherweise nicht installiert ist.

**LiveCycle Designer** wird daraufhin mit einem leeren Zeichenbereich geöffnet. In der Datenansicht ist keine Datenverbindung enthalten, da keine verbundene Message vorhanden ist.

5 Importieren Sie die Formularvorlage, indem Sie **Datei > Vorlage öffnen** auswählen. Das Dialogfeld **Vorlage auswählen** wird geöffnet.

6 Aktivieren Sie das Kontrollkästchen **Aus Design Center-Ressourcenset**, und wählen Sie die Formularvorlagendatei aus. LiveCycle Designer wird mit der im Zeichenbereich angezeigten Formularvorlage gestartet.

7 Speichern Sie Ihre Eingaben, und beenden Sie das Process-Tool.

### So schließen Sie das Projekt ab

1 Konfigurieren Sie die Platform.

2 Konfigurieren Sie die Runtime.

3 Exportieren Sie das Projekt.

**Hinweis:** Beim Konfigurieren der Runtime-Event-Einstellungen können Sie durch Aktivieren der Option **Verbleibende Daten ignorieren** die Leistung verbessern. Bei Verwendung dieser Option werden die verbleibenden Daten ignoriert, wenn das Triggermuster gefunden wurde. Weitere Informationen finden Sie im Abschnitt zur Runtime-Konfiguration des Handbuchs *StreamServe Design Center.*

Im nächsten Schritt wird das Projekt bereitgestellt und ausgeführt (siehe *Bereitstellen und Ausführen* auf Seite 18).

# Bereitstellen und Ausführen

### So stellen Sie das Projekt bereit

Ein Projekt für eine StreamServer-Anwendung stellen Sie in Control Center bereit:

1. Erstellen Sie die StreamServer-Anwendung, für die das Projekt bereitgestellt werden soll.
2. Stellen Sie das Projekt für die StreamServer-Anwendung bereit.

Weitere Informationen hierzu finden Sie in der Dokumentation zu *StreamServe Control Center*.

### So führen Sie die StreamServer-Anwendung aus

Starten Sie die StreamServer-Anwendung in Control Center, indem Sie mit der rechten Maustaste auf den StreamServer-Anwendungsknoten klicken und die Option **Start** wählen.

Das Stoppen und erneute Bereitstellen von StreamServer-Anwendungen erfolgt ebenfalls in Control Center. Weitere Informationen hierzu finden Sie in der Dokumentation zu *StreamServe Control Center*.

# Verwenden einer neu erstellten Formularvorlage

Dieses Kapitel geht auf folgende Arbeitsschritte ein:

- Erstellen einer neuen Formularvorlage unter Verwendung einer StreamServe-Message für Feld- und Blockbindungen.
- Aktivieren der Formularvorlage für den Ausdruck in der Produktionsumgebung.

### Funktionsweise in StreamServer

1. Das Event erkennt die Eingabedatei, extrahiert die Daten in eine Message und löst den Process aus.
2. Der Process führt den Inhalt der Message mit der Formularvorlage zusammen und erzeugt strukturierte Ausgabedaten.
3. Die erzeugten Ausgabedaten werden an einen Treiber und die Nachbearbeitung gesendet.
4. Die formatierte Ausgabe wird an das betreffende Ausgabeziel gesendet.

### In diesem Kapitel behandelte Themen


- *Konfigurieren des Projekts* auf Seite 20.
- *Bereitstellen und Ausführen des Projekts* auf Seite 22.

# Konfigurieren des Projekts

Voraussetzungen:

- Ein in Design Center geöffnetes StreamServe-Projekt.
- Eine Eingabedatei mit feldbasierten Eingabedaten.

### So erstellen Sie die Message

1. Wählen Sie **Datei > Neu > Message**. Es wird eine leere Message erstellt.
2. Benennen Sie die Message um.

### So erstellen Sie das Event

1. Klicken Sie mit der rechten Maustaste auf eine beliebige Stelle in der Message-Ansicht, und wählen Sie **Event hinzufügen > Event-Typ**. Der **Event-Typ** für feldbasierte Eingabedaten ist StreamIN. Der Message-Ansicht wird ein neues Event hinzugefügt.
2. Benennen Sie das Event um.
3. Klicken Sie mit der rechten Maustaste auf das Event, und wählen Sie die Option **Öffnen** aus. Das Event-Tool wird geöffnet.
4. Konfigurieren Sie das Event. Weitere Informationen finden Sie in der Dokumentation zu *StreamServe Design Center.*
5. Speichern Sie Ihre Eingaben, und beenden Sie das Event-Tool.

### So erstellen Sie den Process

1. Klicken Sie mit der rechten Maustaste auf eine beliebige Stelle in der Message-Ansicht, und wählen Sie die Option **Process hinzufügen > Adobe LiveCycle Designer ES2** aus. Der Message-Ansicht wird ein neuer Process hinzugefügt.
2. Benennen Sie den Process um.
3. Klicken Sie mit der rechten Maustaste auf den Process, und wählen Sie die Option **Öffnen** aus. Das Process-Tool wird geöffnet und das Dialogfeld **Einstellungen** angezeigt. Informationen hierzu finden Sie unter *Dialogfeld „Einstellungen“* auf Seite 86.
4. Aktivieren Sie im Dialogfeld **Einstellungen** die Optionen **LiveCycle Designer-Oberfläche beim Start laden** und **Message als Datenverbindung in der Datenansicht hinzufügen**.
5. Klicken Sie auf **OK**. Das Dialogfeld **Einstellungen** wird geschlossen, und im anschließend geöffneten LiveCycle Designer werden ein leerer Zeichenbereich und der Inhalt der Message-Struktur als Datenverbindung angezeigt.
6. Konfigurieren Sie den Process, und verschieben Sie per Drag & Drop Felder aus der Message in den Zeichenbereich. Für jedes Feld wird ein Formularobjekt mit einer Bindung an das jeweilige Feld erstellt.

7 Wählen Sie **Speichern**. Beim erstmaligen Speichern wird das Dialogfeld **Ressource zum Speichern der XDP-Hauptvorlage auswählen** geöffnet.

8 Wählen Sie die Ressource aus, und klicken Sie auf **OK**.

### So schließen Sie das Projekt ab

1 Konfigurieren Sie die Platform.

2 Konfigurieren Sie die Runtime.

3 Exportieren Sie das Projekt.

Weitere Informationen finden Sie in der Dokumentation zu *StreamServe Design Center*.

Im nächsten Schritt wird das Projekt bereitgestellt und ausgeführt (siehe *Bereitstellen und Ausführen des Projekts* auf Seite 22).

# Bereitstellen und Ausführen des Projekts

### So stellen Sie das Projekt bereit

Ein Projekt für eine StreamServer-Anwendung stellen Sie in Control Center bereit:

1. Erstellen Sie die StreamServer-Anwendung, für die das Projekt bereitgestellt werden soll.
2. Stellen Sie das Projekt für die StreamServer-Anwendung bereit.

Weitere Informationen hierzu finden Sie in der Dokumentation zu *StreamServe Control Center*.

### So führen Sie die StreamServer-Anwendung aus

Starten Sie die StreamServer-Anwendung in Control Center, indem Sie mit der rechten Maustaste auf den StreamServer-Anwendungsknoten klicken und die Option **Start** wählen.

Das Stoppen und erneute Bereitstellen von StreamServer-Anwendungen erfolgt ebenfalls in Control Center. Weitere Informationen hierzu finden Sie in der Dokumentation zu *StreamServe Control Center*.

# Verwenden von Vorlagen

**Hinweis:** Das dynamische Laden von Vorlagen kann sich negativ auf die Leistung auswirken. Informationen hierzu finden Sie unter *Überlegungen zur Leistung* auf Seite 12.

### Verweise in dynamisch geladenen Vorlagen

Vorlagen mit Abhängigkeiten (externen Verweisen auf Fragmente und Bilder) werden unterstützt, wenn der Process auf folgendem Wege auf die referenzierten Dateien zugreifen kann:

- statische Pfade
- relative Pfade in Bezug auf den Formularvorlagenordner im Dateisystem
- HTTP-URL

### Verweise in Vorlagen aus dem LiveCycle-Repository

Vorlagen, die dynamisch aus dem LiveCycle ES2-Repository geladen werden, dürfen keine Verweise auf andere Ressourcen im Repository enthalten. Damit müssen jegliche Bilder eingebettet bzw. über HTTP, Dateipfade, FTP-URIs referenziert sein.

### Anwendungsversion

Es ist möglich, dass mehrere Versionen einer Ressource im LiveCycle ES2-Repository mit unterschiedlichen LiveCycle-Anwendungsversionen verknüpft sind. Beachten Sie, dass LiveCycle Production Print ES2 nicht automatisch die letzte Anwendungsversion verwendet.

### In diesem Kapitel behandelte Themen


- *Verwenden einer StreamServe-Variablen zum Laden der Vorlage* auf Seite 24
- *Verwenden eines SOM-Ausdrucks zum Laden einer Vorlage* auf Seite 26
- *Laden einer Vorlage direkt aus dem LiveCycle-Repository* auf Seite 27
- *Laden von Schriftarten* auf Seite 28

# Verwenden einer StreamServe-Variablen zum Laden der Vorlage

Die Verwendung einer StreamServe-Variablen ist sinnvoll, wenn der Vorlagenpfad nicht durch einen SOM-Ausdruck angegeben werden kann oder mithilfe von Skriptcode in StreamServe angegeben werden muss.

Die Variable wird direkt nach dem Starten des Process für jeden Job einmal ausgewertet. Das bedeutet, dass die Vorlage für alle im Rahmen des Jobs verarbeiteten Datensätze verwendet wird.

Die Variable steht für eine Zeichenfolge, die sowohl Dateipfade als auch URIs (File-, HTTP- und Repository-URIs) enthalten kann. Pfade und URIs können absolut oder relativ (zum StreamServer-Arbeitsverzeichnis) angegeben werden.

Bei HTTP-URIs kann die einfache HTTP-Authentifizierung verwendet werden.

*Beispiel 1* *Beispiele für Dateipfade und URIs*

- Dateipfad:
  `C:\templates\mytemplate.xdp`
  oder
  `../../MyTemplate.xdp`
- File-URI:
  `file:///D:/my%20templates/mytemplate.xdp`
- HTTP-URI:
  `HTTP://www.exampletemplate.com`
- Repository-URI:
  `LC://myhost:8080/myfolder/myresource.xdp`

*Beispiel 2* *Die Auswahl der Vorlage erfolgt auf Grundlage externer Daten*

Die für eine bestimmte Benutzerkategorie verwendete Vorlagendatei wird auf der Festplatte (oder in einer Datenbank) gespeichert.

Mithilfe eines Vor-Process-StreamServe-Skripts wird basierend auf der Benutzerkategorie ein Wert aus der Datei gelesen, und der Vorlagenpfad wird der Variablen zugewiesen. Die Variable wird dann zur Vorlagenauswahl im Process verwendet.

Beispiel für absoluten Pfad:

```
$template =„C:\templates\dynamic_invoice.xdp“;
```

Beispiel für relativen Pfad (bezogen auf das Arbeitsverzeichnis):

```
$template = „../data/XDP_template/dynamic_invoice.xdp“;
```

*Beispiel 3* *Vorlagenauswahl basierend auf Metadaten*

Der Vorlagenpfad wird als ein HTTP-Headerwert an StreamServe gesendet.

Ein Skript extrahiert den Wert aus dem HTTP-Header und weist ihn einer Variablen zu. Die Variable wird dann zur Vorlagenauswahl im Process verwendet.

### So können Sie zur Laufzeit mithilfe einer StreamServe-Variablen eine Vorlage laden

1. Klicken Sie im Process-Tool auf **Datei > Vorlage öffnen**. Das *Dialogfeld „Vorlage auswählen“* wird geöffnet.
2. Wählen Sie **StreamServe-Variable**, und geben Sie den Namen der Variablen ein, die auf die Vorlage verweist.
3. Wenn Sie eine Verbindung zu einem LiveCycle-Repository herstellen möchten, geben Sie die entsprechenden Details unter **Verbindung zum Runtime-Repository** ein.
4. Bei Verwendung einer HTTP-URI mit einfacher HTTP-Authentifizierung wählen Sie **Einfache HTTP-Authentifizierung verwenden**, und geben Sie die entsprechenden Anmeldeinformationen ein.
5. Klicken Sie auf **OK**. Das Startfenster des Process-Tools wird geöffnet.

# Verwenden eines SOM-Ausdrucks zum Laden einer Vorlage

### Verwenden des SOM-Ausdrucks

Der SOM-Ausdruck ist sinnvoll, wenn auf den Pfad zu einer Vorlage über das Daten-DOM zugegriffen werden kann und wenn Sie für jeden Datensatz in einem Batch-Job eine andere Vorlage benötigen.

Der SOM-Ausdruck kann einmal pro Datensatz oder einmal für jede verarbeitete Seite ausgewertet werden. Daher können für jeden zu verarbeitenden Datensatz (bzw. jede Seite) andere Vorlagen verwendet werden.

Dateipfade und URIs werden auf dieselbe Weise wie bei einer StreamServe-Variablen unterstützt (siehe *Verwenden einer StreamServe-Variablen zum Laden der Vorlage* auf Seite 24).

*Beispiel 4* *SOM-Ausdruck*

Der Vorlagenpfad ist in einem Element in der Eingabedatei enthalten.

Ein auf das Element verweisender SOM-Ausdruck wird für die Vorlagenauswahl im Process verwendet.

### So können Sie zur Laufzeit mithilfe eines SOM-Ausdrucks eine Vorlage laden

1. Klicken Sie im Process-Tool auf **Datei > Vorlage öffnen**. Das *Dialogfeld „Vorlage auswählen“* wird geöffnet.
2. Wählen Sie **SOM-Ausdruck**, und geben Sie einen SOM-Ausdruck in das Daten-DOM ein, das auf die Vorlage verweist.
3. Wenn Sie eine Verbindung zu einem LiveCycle-Repository herstellen möchten, geben Sie die entsprechenden Details unter **Verbindung zum Runtime-Repository** ein.
4. Bei Verwendung einer HTTP-URI mit einfacher HTTP-Authentifizierung wählen Sie **Einfache HTTP-Authentifizierung verwenden**, und geben Sie die entsprechenden Anmeldeinformationen ein.
5. Klicken Sie auf **OK**. Das Startfenster des Process-Tools wird geöffnet.

# Laden einer Vorlage direkt aus dem LiveCycle-Repository

Vorlagen können zur Entwurfszeit statisch direkt aus dem LiveCycle-Repository geladen werden. StreamServer ruft die Vorlagen dann zur Laufzeit aus dem Repository ab. Damit hat der Benutzer zur Laufzeit bei Bedarf die Möglichkeit, eine Vorlage zu aktualisieren und eine neue Version im Repository abzulegen. Der Server lädt dann jeweils die letzte gespeicherte Vorlage.

### So laden Sie eine Vorlage aus dem LiveCycle-Repository

1. Klicken Sie im Process-Tool auf **Datei > Vorlage öffnen**. Das *Dialogfeld „Vorlage auswählen“* wird geöffnet.
2. Geben Sie unter **Verbindung zum Runtime-Repository** die erforderlichen Details für das LiveCycle-Repository ein.
3. Wählen Sie die Option **Aus dem LiveCycle-Repository**, und navigieren Sie zu der gewünschten Vorlage im Repository. Über die angegebenen Verbindungsdetails wird die Verbindung zum Repository hergestellt.
4. Klicken Sie auf **OK**. Das Startfenster des Process-Tools wird geöffnet.

# Laden von Schriftarten

Aus Leistungsgründen lädt StreamServe alle Schriftarten während des Programmstarts von StreamServer.

Bei Verwendung statisch geladener Vorlagen enthält das Design Center-Exportpaket alle referenzierten Schriftarten.

Bei Verwendung dynamisch geladener Vorlagen müssen Sie manuell die Schriftarten einbinden, die von den dynamischen Vorlagen genutzt werden.

Das bedeutet, Sie müssen die Schriftarten dem Ressourcenset in einem Projekt hinzufügen.

### So fügen Sie die Schriftarten dem Ressourcenset in einem Projekt hinzu

Importieren Sie die Schriftarten manuell in das Projekt, wie in der Dokumentation zu *Design Center* beschrieben.

### Tipp für alternative Vorgehensweise

Andere Möglichkeit zum manuellen Einbinden der Schriftarten: Importieren Sie eine oder mehrere XDP-Vorlagen, die alle als Ressourcen verwendeten Schriftarten enthalten, und verbinden Sie sie einzeln mit einem (LiveCycle-)Process-Tool.

Speichern Sie den Process, nachdem Sie alle XDPs hinzugefügt haben. Daraufhin werden automatisch alle verwendeten Schriftarten in das Projekt importiert.

Anschließend können Sie die XDP-Ressourcen wieder entfernen.
Weitere Informationen finden Sie in der Dokumentation zu *Design Center*.

# Arbeiten mit externen Vorlagen

Sie können mit einer aus einer StreamServe-Message exportierten XML-Schemadatei eine Vorlage in einer eigenständigen LiveCycle Designer ES2-Instanz bearbeiten und entwerfen. Die exportierte Datei nutzt die StreamServe-Message in der Datenansicht als Datenverbindung. Die Vorlage kann in die StreamServe-Lösung erneut importiert werden. Das ist beispielsweise zweckmäßig, wenn Sie die Entwicklung und Wartung von Vorlagen auslagern.

Sie können eine Vorschau des XML-Schemas mit Musterdaten erstellen. Die Musterdaten für jedes Feld können im StreamServe-Event-Tool, z. B. PageIN oder XMLIN, eingegeben werden.

Dies ist besonders dann hilfreich, wenn die Eingabedaten nicht im XML-Format vorliegen, z. B. als ASCII-Text. LiveCycle Production Print ES2 wandelt die Eingabedaten intern in XML um. Wenn Sie jedoch in LiveCycle Designer ES2 eine Vorschau erstellen möchten, liegt meistens noch keine XML-Datei vor, die für die Vorschau verwendet werden kann. Mit der Exportvorschaufunktion für XML können Sie in diesem Szenario eine XML-Datei erstellen.

### Verwendungsszenario

Ein mögliches Szenario wäre, dass Sie den Entwurf der Vorlage an einen externen Designer vergeben haben. Sie erstellen zwei Exportdateien, eine Message-Schemadatei und eine Vorschaudatei mit Musterdaten, und übergeben diese an den externen Designer. Wenn der externe Designer seine Arbeit beendet hat, importieren Sie die Vorlage wieder in Design Center.

### So exportieren Sie eine StreamServe-Message

1. Klicken Sie im Process-Tool auf **Datei > Message-Schema exportieren**. Das Dialogfeld **Speichern unter** wird geöffnet.
2. Wählen Sie den Speicherort aus, und weisen Sie der Schemadatei (XSD) einen Namen zu.

### So erstellen Sie eine XML-Vorschaudatei mit Musterdaten

1. Öffnen Sie das Dialogfeld **Einstellungen**, und überprüfen Sie, ob die Option **Message als Datenverbindung in der Datenansicht hinzufügen** aktiviert ist.
2. Erstellen Sie eine XML-Vorschaudatei mit Musterdaten, und wählen Sie **Datei > XML-Vorschau exportieren**.
3. Wählen Sie den Speicherort aus, und weisen Sie der Schemadatei (XSD) einen Namen zu.

### So importieren Sie eine exportierte Vorlage wieder in das Design Center-Ressourcenset

1. Öffnen Sie die Ressourcenset-Ansicht.
2. Klicken Sie auf **Importieren**, und wählen Sie die zu importierende Vorlagendatei aus. Daraufhin wird das Dialogfeld **Ressourcentyp-einstellungen** geöffnet.
3. Wählen Sie aus der Dropdownliste den Ressourcentyp **XDP-Vorlage**, und klicken Sie auf **OK**. Daraufhin wird dem Ressourcenset die neue Ressource hinzugefügt.

# Integration mit LiveCycle Production Print ES2 und LiveCycle ES2

### Zugriff von StreamServe Design Center auf ein LiveCycle ES2-Repository

StreamServe Design Center kann mit einem LiveCycle ES2-Repository verbunden werden. Auf diese Weise können Vorlagen und andere zugehörige Ressourcen mit LiveCycle Production Print ES2 verwendet werden, ohne sie über das Dateisystem importieren zu müssen. Die Vorlagen und dazugehörigen Dateien werden aus dem LiveCycle ES2-Repository in ein Ressourcenset im Design Center-Projekt importiert.

### Aufrufen von LiveCycle Production Print ES2-Anwendungen durch LiveCycle ES2

LiveCycle ES2 kann StreamServer-Anwendungen aufrufen, die durch Webdienste offengelegt werden. Mithilfe dieser Webdienste können bei der Dokumentverarbeitung StreamServer-Anwendungen in LiveCycle ES2-Prozesse integriert werden.

### Aufrufen von LiveCycle-Prozessen durch LiveCycle Production Print ES2

StreamServer kann LiveCycle-Prozesse aufrufen, die in LiveCycle ES2 bereitgestellt und über Webdienste offengelegt werden. Mithilfe dieser Webdienste können bei der Dokumentverarbeitung LiveCycle-Prozesse in die StreamServer-Pipeline integriert werden.

### In diesem Kapitel behandelte Themen


- *Zugriff auf das Adobe LiveCycle ES2-Repository* auf Seite 32
- *Aufrufen von LiveCycle Production Print ES2 aus LiveCycle ES2* auf Seite 37
- *Aufrufen von LiveCycle ES2-Prozessen aus LiveCycle Production Print ES2* auf Seite 45

# Zugriff auf das Adobe LiveCycle ES2-Repository

Vorlagen und zugehörige Dateien können aus dem LiveCycle ES2-Repository nach Design Center importiert und als Ressourcensets gespeichert werden.

Der Zugriff auf das LiveCycle ES2-Repository aus Design Center bietet folgende Möglichkeiten:

- Durchsuchen des LiveCycle-Repository. Im Browser werden folgende Informationen angezeigt:
  - Ob die Ressource bereits lokal im Design Center-Ressourcenset vorhanden ist.
  - Ob sie im Repository aktualisiert wurde und lokal im Design Center-Ressourcenset aktualisiert werden muss.
  - Ob sie lokal im Design Center-Ressourcenset aktualisiert wurde.
  - Ob sie aus dem Repository entfernt wurde.

  Informationen hierzu finden Sie unter *Beim Zugriff auf das LiveCycle ES2-Repository verwendete Symbole* auf Seite 99.
- Importieren von Ressourcen und deren Abhängigkeiten aus dem LiveCycle ES2-Repository.
- Aktualisieren bereits aus dem LiveCycle ES2-Repository importierter Ressourcen.

Da Sie nur Schreibzugriff haben, können Sie beim Zugriff aus Design Center keine Daten im LiveCycle ES2-Repository ändern.

**Hinweis:** Sie können Verbindungen zu verschiedenen Repositorys definieren, jedoch jeweils nur Ressourcen für ein Repository erstellen und aktualisieren.

## Herstellen von Verbindungen zu LiveCycle-Repositorys

Sie können in Design Center Verbindungen zu einem LiveCycle ES2-Repository definieren und aktivieren. Dabei können Sie jeweils eine Verbindung als aktive Verbindung festlegen. Die Verbindung ist nur während der Kommunikation (während des Imports und der Aktualisierung) aktiv.

Wählen Sie **Extras > Adobe LiveCycle-Repositoryverbindung auswählen**. Das *Dialogfeld „Aktive Adobe LiveCycle-Repositoryverbindung auswählen“* wird geöffnet.

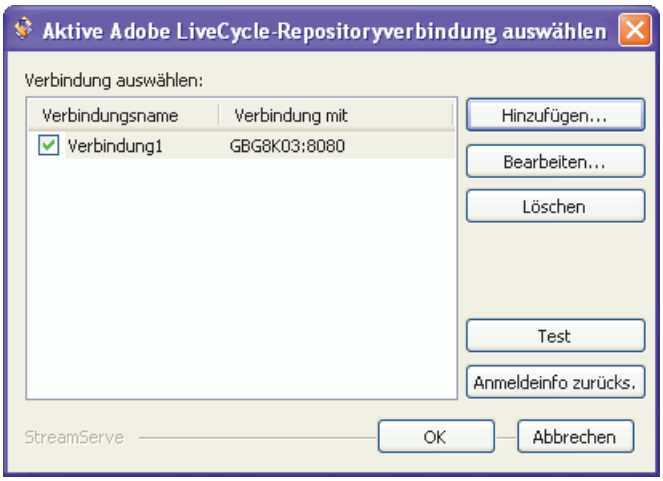



*Abbildung 1 Dialogfeld „Aktive Adobe LiveCycle-Repositoryverbindung auswählen“*

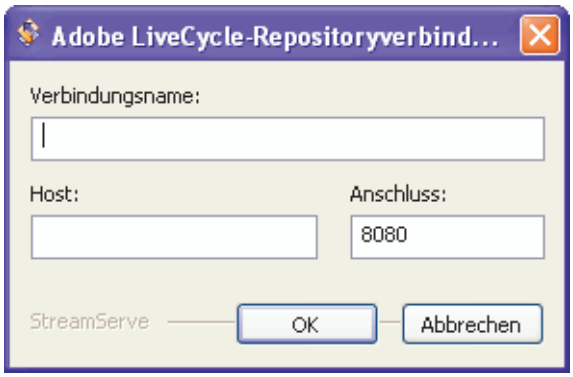
### So aktivieren Sie eine Verbindung

Aktivieren Sie das Kontrollkästchen für die zu aktivierende Verbindung. Die ausgewählte Verbindung wird bei Bedarf (d. h. während des Imports aus dem Repository und der damit verbundenen Aktualisierung) aktiviert.

### So fügen Sie eine Verbindung hinzu bzw. bearbeiten sie

1 Klicken Sie auf **Hinzufügen** oder **Bearbeiten**. Das Dialogfeld **Adobe LiveCycle-Repositoryverbindung bearbeiten** wird geöffnet.

2 Legen Sie die Einstellungen fest.




- **Verbindungsname**: Wählen Sie einen geeigneten Namen für die Verbindung aus.
- **Host**: Hostname oder IP-Adresse des Servers, auf dem sich das Repository befindet.
- **Anschluss**: Der für die Kommunikation mit dem Host verwendete Anschluss.

3 Klicken Sie auf **OK**. Die Verbindung muss vorhanden und verfügbar sein, da sie beim Hinzufügen hergestellt und identifiziert wird. Wenn Sie nicht vorhanden ist, wird eine Fehlermeldung angezeigt.

### So löschen Sie eine Verbindung

1 Wählen Sie die zu löschende Verbindung aus.

2 Klicken Sie auf **Löschen**.

### So testen Sie eine Verbindung

Sie können überprüfen, ob eine Verbindung zu einem Adobe LiveCycle-Repository funktioniert.

1 Wählen Sie die zu testende Verbindung aus.

2 Klicken Sie auf **Test**. Sie werden aufgefordert, die Anmeldeinformationen einzugeben.

3 Geben Sie Ihre Anmeldeinformationen ein, und klicken Sie auf **OK**.

## Importieren einer Ressource aus dem Adobe LiveCycle ES2-Repository

Beim Importieren einer Ressource aus dem LiveCycle ES2-Repository wird eine lokale Kopie der Ressource im Design Center-Ressourcenset erstellt.

Die Ressource wird dann in einer Pfad- und Dateistruktur hinzugefügt, die der Struktur des LiveCycle ES2-Repository entspricht.

**Hinweis:** Ändern Sie keinesfalls die Struktur des Design Center-Ressourcensets. Verweise auf Fragmente und Bilder können möglicherweise verloren gehen, wenn deren interne relative Positionen geändert werden.

Sie können eine Ressource mit oder ohne Abhängigkeiten importieren.

### So importieren Sie eine Ressource

1 Wählen Sie **Ressourcen > Aus Adobe LiveCycle-Repository importieren**. Das Dialogfeld **Ressource auswählen** wird geöffnet.

2 Suchen Sie die zu importierende Ressource, und doppelklicken Sie darauf. Das *Dialogfeld „Adobe LiveCycle-Repositoryressource importieren“* wird geöffnet.

Die Option **Grundsätzlich Kopfversion aller Ressourcen auschecken** ist standardmäßig aktiviert. Das bedeutet, dass die aktuelle Version der Ressource und deren Abhängigkeiten importiert werden.

3 Wenn Sie eine andere Version als die Kopfversion importieren möchten, deaktivieren Sie das Kontrollkästchen, und wählen Sie eine Version aus der Dropdownliste aus.

4 Wählen Sie die zu importierende Ressource und deren Abhängigkeiten aus, und klicken Sie auf **OK**.

# Aktualisieren von Ressourcen aus dem Adobe ES2 LiveCycle-Repository

### So aktualisieren Sie eine Ressource und deren Abhängigkeiten

Wenn die ursprüngliche Ressource (im LiveCycle ES2-Repository) geändert wurde, können Sie die lokale Kopie im Design Center-Ressourcenset aktualisieren.

1. Klicken Sie in der Ressourcenset-Ansicht mit der rechten Maustaste auf die Ressource, und wählen Sie die Option **Von Ursprung aktualisieren** aus. Das *Dialogfeld „Alle Adobe LiveCycle-Repositoryressourcen aktualisieren“* wird geöffnet. In diesem Dialogfeld werden der Status der lokalen Kopie und deren Abhängigkeiten im Vergleich mit der Ressource im Repository angezeigt. Weitere Informationen hierzu finden Sie unter *Beim Zugriff auf das LiveCycle ES2-Repository verwendete Symbole* auf Seite 99.
2. Wählen Sie die zu importierende Ressource und deren Abhängigkeiten aus.
3. Klicken Sie auf **OK**. Die Ressource und die ausgewählten Abhängigkeiten werden in derselben Pfad- und Dateistruktur wie im Adobe LiveCycle-Repository gespeichert.

### So aktualisieren Sie mehrere Ressourcen

Wenn sich im LiveCycle ES2-Repository eine Vielzahl von Ressourcen geändert hat, können Sie mehrere Ressourcen auf einmal aktualisieren.

1. Wählen Sie **Ressourcen > Alle Adobe LiveCycle-Ressourcen aktualisieren**. Das *Dialogfeld „Alle Adobe LiveCycle-Repositoryressourcen aktualisieren“* wird geöffnet. In diesem Dialogfeld werden die importierten Ressourcen und deren Status im Vergleich mit den Repositoryressourcen angezeigt. Informationen hierzu finden Sie unter *Beim Zugriff auf das LiveCycle ES2-Repository verwendete Symbole* auf Seite 99.

**Hinweis:** Die Abhängigkeiten werden bei Verwendung des Befehls **Alle Adobe LiveCycle-Ressourcen aktualisieren** nicht aktualisiert.

2. Wählen Sie die zu aktualisierenden Ressourcen über die zugehörigen Kontrollkästchen aus, oder wählen Sie **Alle auswählen**.
3. Klicken Sie auf **OK**. Die ausgewählten Ressourcen werden in der gleichen Pfad- und Dateistruktur wie im LiveCycle ES2-Repository gespeichert.

# Aufrufen von LiveCycle Production Print ES2 aus LiveCycle ES2

LiveCycle ES2 kann StreamServer-Anwendungen aufrufen, die durch Webdienste offengelegt werden. Mithilfe dieser Webdienste können bei der Dokumentverarbeitung StreamServer-Anwendungen in LiveCycle ES2-Prozesse integriert werden.

### Eingabe- und Ausgabedaten

Der StreamServer-Dienst ist vollkommen generisch. Es können Daten beliebigen Typs an StreamServer gesendet werden, z. B. eine XML-Datendatei, die mit einer Formularvorlage zusammengeführt werden soll.

Auch das Ergebnis der StreamServer-Verarbeitung ist generisch. Je nach StreamServer-Konfiguration können die unterschiedlichsten Daten, von einer Druckdatei bis hin zu einer Statusmeldung, ausgegeben werden.

### Für die Integration verwendete Document Service Component

Die Document Service Component (DSC) mit der Bezeichnung LiveCycle Production Print DSC wurde für diesen Zweck entwickelt. Diese DSC kann in jedem LiveCycle ES2-Prozess zur Übertragung der Daten von und nach StreamServer verwendet werden.

Die LiveCycle Production Print DSC finden Sie auf dem Installationsdatenträger als JAR-Datei (`lcppdsc.jar`). Sie können die DSC über Workbench in LiveCycle ES2 bereitstellen.

### Für die Integration verwendete StreamServer-Connectors

Die Webdienste werden von StreamServer mithilfe von Dienstanforderungs-Eingabe-Connectors offengelegt. StreamServer empfängt den Job über den Dienstanforderungs-Connector von LiveCycle ES2 und kann die verarbeiteten Ausgabedaten über einen beliebigen Ausgabe-Connector an LiveCycle ES2 zurückgeben.

### Dienst-Gateway

Es muss ein Dienst-Gateway ausgeführt werden. Das Dienst-Gateway verwaltet die Webdienstaufrufe zwischen LiveCycle ES2 und StreamServer.

### Musterprojekt

Ein Design Center-Projekt mit dem Namen `sampleproject.dcpackage` steht auf dem Installationsmedium zur Verfügung. Das Projekt stellt diesen Einsatz musterhaft dar. Es veranschaulicht sowohl die Vorgänge auf der LiveCycle ES2-Prozessseite als auch auf der LiveCycle Production Print-Seite. Es ist ratsam, dieses Musterprojekt genau zu studieren und auszuprobieren. Das Musterprojekt ist auch hilfreich als Vorlage bei der Entwicklung eigener integrierter Prozesse. Weitere Informationen zum Musterprojekt finden Sie unter *Das Musterprojekt* auf Seite 59.

# Merkmale der LiveCycle Production Print DSC

Der LiveCycle Production Print ES2-Workbench-Dienst hat drei Funktionen:

- *Post* – Senden eines Jobs von LiveCycle ES2 an StreamServer. Es werden keine Statusinformationen zurückgegeben.
- *Run* – Senden eines Jobs von LiveCycle ES2 an StreamServer. Statusinformationen werden nach Abschluss des Ausgabe-Jobs durch StreamServer zurückgegeben.
- *Generate* – Senden eines Jobs von LiveCycle ES2 an StreamServer zur Verarbeitung und anschließender Empfang des verarbeiteten Jobs sowie der Statusinformationen von StreamServer.

## Post

Verwenden Sie diese Funktion, wenn LiveCycle ES2 nur einen Job zur Verarbeitung an StreamServer senden soll und keine Statusinformationen nach Abschluss des Ausgabe-Jobs durch StreamServer erforderlich sind.

### Parameter

Für diese Funktion können die folgenden Parameter verwendet werden.

| Parameter | Typ und untergeordneter Typ | Beschreibung |
|---|---|---|
| **Remoteendpunkt** | **Typ**: Zeichenfolge | Die Adresse des Dienst-Gateways. Beispiel: `http://localhost:2718` |
| **Remotedienstname** | **Typ**: Zeichenfolge | Der Name (Groß-/Kleinschreibung wird unterschieden) des aufzurufenden StreamServer-Dienstes. Muss genau dem `Dienstnamen` des Dienstanforderungs-Eingabe-Connectors entsprechen, der von StreamServer zum Abrufen des Jobs verwendet wird. Informationen hierzu finden Sie unter *Erstellen eines Dienstanforderungs-Eingabe-Connectors* auf Seite 43. |
| **Zeitlimit für Verbindungen** | **Typ**: Ganzzahl | Die maximale Zeitdauer (in Sekunden), in der StreamServer den Job abrufen kann. Ein Zeitlimit <=0 bedeutet, dass es kein Zeitlimit gibt. |
| **Eingabedaten** | **Typ**: Document | Der Job (Dokument, Stapelausführung usw.) und Inhaltstyp des von StreamServer zu verarbeitenden Jobs. |
| **Vorlage** | **Typ**: Document | XDP wird von StreamServer bei der Job-Verarbeitung verwendet. |

| Parameter | Typ und untergeordneter Typ | Beschreibung |
|---|---|---|
| **Zusätzliche Eingabeparameter** | **Typ**: Liste<br>**Untergeordneter Typ**: Eingabeparameter | An StreamServer zu übergebende Variablen. StreamServer kann nur mithilfe der Funktion `GetConnectorValue` auf diese Variablen zugreifen. |

## Run

Verwenden Sie diese Funktion, wenn LiveCycle ES2 einen Job zur Verarbeitung an StreamServer senden soll und StreamServer nach Abschluss des Ausgabe-Jobs Statusinformationen zurückgeben soll.

### Parameter

Für diese Funktion können die folgenden Parameter verwendet werden.

| Parameter | Typ und untergeordneter Typ | Beschreibung |
|---|---|---|
| **Remoteendpunkt** | **Typ**: Zeichenfolge | Die Adresse des Dienst-Gateways. Beispiel:<br>`http://localhost:2718` |
| **Remotedienstname** | **Typ**: Zeichenfolge | Der Name (Groß-/Kleinschreibung wird unterschieden) des aufzurufenden StreamServer-Dienstes. Muss genau dem `Dienstnamen` des Dienstanforderungs-Eingabe-Connectors entsprechen, der von StreamServer zum Abrufen des Jobs verwendet wird. Informationen hierzu finden Sie unter *Erstellen eines Dienstanforderungs-Eingabe-Connectors* auf Seite 43. |
| **Zeitlimit für Verbindungen** | **Typ**: Ganzzahl | Die maximale Zeitdauer (in Sekunden), in der StreamServer antworten muss. Wenn StreamServer bei der Job-Verarbeitung das Zeitlimit überschreitet, wird der Job nicht aus der Warteschlangendatenbank entfernt. Ein Zeitlimit <=0 bedeutet, dass es kein Zeitlimit gibt. |
| **Eingabedaten** | **Typ**: Document | Der Job (Dokument, Stapelausführung usw.) und Inhaltstyp des von StreamServer zu verarbeitenden Jobs. |
| **Vorlage** | **Typ**: Document | XDP wird von StreamServer bei der Job-Verarbeitung verwendet. |
| **Zusätzliche Eingabeparameter** | **Typ**: Liste<br>**Untergeordneter Typ**: Eingabeparameter | An StreamServer zu übergebende Variablen. StreamServer kann nur mithilfe der Skriptfunktion `GetConnectorValue` auf diese Variablen zugreifen. |

| Parameter | Typ und untergeordneter Typ | Beschreibung |
|---|---|---|
| **Zurückgegebene Status und Dokumente** | **Typ**: Ergebnisstatus und Dokument(e) | Von StreamServer zurückgegebene Status und Dokumente. |
| **Unterparameter für zurückgegebene Status und Dokumente** | | |
| **statusCode** | **Typ**: Ganzzahl | Von StreamServer zurückgegebener Statuscode.<br>0: OK<br>1: Warnung |
| **statusMessage** | **Typ**: Zeichenfolge | Zusätzliche Statusinformationen. Enthält detaillierte Informationen zu dem von StreamServer zurückgegebenen statusCode. |

## Generate

Verwenden Sie diese Funktion, wenn LiveCycle ES2 einen Job an StreamServer zur Verarbeitung senden und dann den verarbeiteten Job wieder von StreamServer abrufen soll.

### Parameter

Für diese Funktion können die folgenden Parameter verwendet werden.

| Parameter | Typ und untergeordneter Typ | Beschreibung |
|---|---|---|
| **Remote-endpunkt** | **Typ**: Zeichenfolge | Die Adresse des Dienst-Gateways. Beispiel:<br>`http://localhost:2718` |
| **Remote-dienstname** | **Typ**: Zeichenfolge | Der Name (Groß-/Kleinschreibung wird unterschieden) des aufzurufenden StreamServer-Dienstes. Muss genau dem `Dienstnamen` des Dienstanforderungs-Eingabe-Connectors entsprechen, der von StreamServer zum Abrufen des Jobs verwendet wird. Informationen hierzu finden Sie unter *Erstellen eines Dienstanforderungs-Eingabe-Connectors* auf Seite 43. |
| **Zeitlimit für Verbindungen** | **Typ**: Ganzzahl | Die maximale Zeitdauer (in Sekunden), in der StreamServer antworten muss. Wenn StreamServer bei der Job-Verarbeitung das Zeitlimit überschreitet, wird der Job nicht aus der Warteschlangendatenbank entfernt. Ein Zeitlimit <=0 bedeutet, dass es kein Zeitlimit gibt. |

| Parameter | Typ und untergeordneter Typ | Beschreibung |
|---|---|---|
| **Eingabedaten** | **Typ**: Document | Der Job (Dokument, Stapelausführung usw.) und Inhaltstyp des von StreamServer zu verarbeitenden Jobs. |
| **Vorlage** | **Typ**: Document | XDP wird von StreamServer bei der Job-Verarbeitung verwendet. |
| **Zusätzliche Eingabeparameter** | **Typ**: Liste<br>**Untergeordneter Typ**: Eingabeparameter | An StreamServer zu übergebende Variablen. StreamServer kann nur mithilfe der Skriptfunktion `GetConnectorValue` auf diese Variablen zugreifen. |
| **Zurückgegebene Status und Dokumente** | **Typ**: Ergebnisstatus und Dokument(e) | Von StreamServer zurückgegebene Status und Dokumente. |
| **Unterparameter für zurückgegebene Status und Dokumente** | | |
| **statusCode** | **Typ**: Ganzzahl | Von StreamServer zurückgegebener Statuscode.<br>0: OK<br>1: Warnung |
| **statusMessage** | **Typ**: Zeichenfolge | Zusätzliche Statusinformationen. Enthält detaillierte Informationen zu dem von StreamServer zurückgegebenen statusCode. |
| **documents** | **Typ**: Liste<br>**Untergeordneter Typ**: Dokument | Der von StreamServer zurückgegebene Job und Inhaltstyp des Jobs. |

## Fehlerbehandlung

Im Falle eines Fehlers wird die Ausnahme `ProductionPrintException` ausgelöst. Die Fehlercodes werden in der folgenden Tabelle beschrieben.

| Code | Beschreibung |
|---|---|
| `-1` | Dienstserverfehler.<br>Dies ist ein Fehler, den ein Client durch einen erneuten Versuch nicht einfach beheben kann. Gründe können beispielsweise ein Mangel an Arbeitsspeicher oder Festplattenspeicherplatz auf dem Server sein. |
| `-2` | Clientfehler.<br>Dieser Fehler kann von einem Client behoben werden. Gründe können beispielsweise ein ungültiger Dienstname oder ein zu kurzes Zeitintervall sein. |

| Code | Beschreibung |
|---|---|
| -3 | Aufruffehler.<br>Ein ungültiger SOAP-Umschlag wurde an den Webdienst gesendet. |
| -4 | Remotefehler.<br>Beispielsweise ein Netzwerkfehler oder ein falsch angegebener Endpunkt. |
| -5 | Ausgabedatenfehler.<br>Der Client konnte keine Ausgabedaten vom Server empfangen. |
| -6 | Adressierungsfehler.<br>Für den Webdienst wurde ein ungültiger Endpunkt angegeben. |
| -7 | Lokaler E/A-Fehler.<br>Ein DSC-seitiger E/A-Fehler. Beispielsweise nicht genügend Festplattenspeicherplatz auf dem DSC-Host. |
| -8 | Allgemeiner DSC-Fehler.<br>Die Fehlermeldung zur Ausnahme enthält weitere Einzelheiten. |

# StreamServer-Konfiguration

Die StreamServer-Konfiguration umfasst einen Dienstanforderungs-Eingabe-Connector und das entsprechende Event, den Process, den Ausgabe-Connector und Warteschlangen.

### Dienstanforderungs-Eingabe-Connector

Dieser Connector legt den Webdienst für LiveCycle ES2 offen und ruft den Job von LiveCycle ES2 ab.

### Event und Process

Event und Processes werden entsprechend den Design Center-Standardvorgehensweisen konfiguriert.

### Ausgabe-Connector

In einem Szenario, in dem StreamServer die endgültige Ausgabe bereitstellt, wird der Ausgabe-Connector entsprechend den Design Center-Standardvorgehensweisen konfiguriert.

In einem Szenario, in dem LiveCycle ES2 die endgültige Ausgabe bereitstellt, muss der Ausgabe-Connector ebenfalls so konfiguriert werden, dass der Job als Webdienstantwort zurückgegeben wird.

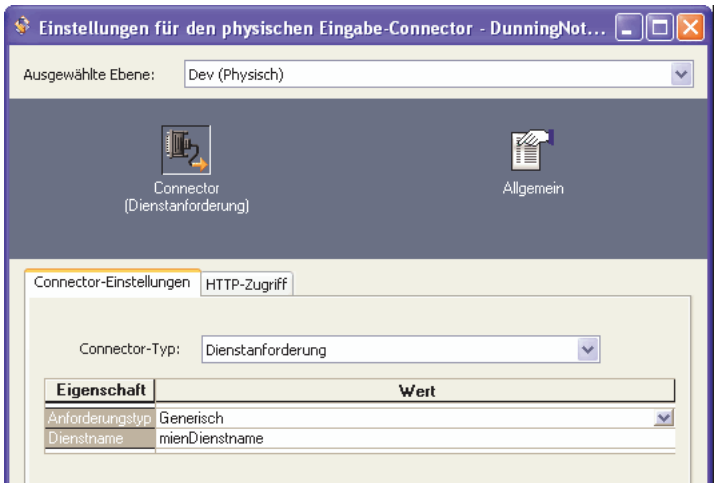
### Warteschlangen

Der Eingabe- und der Ausgabe-Connector müssen mit Warteschlangen verbunden werden.

## Erstellen eines Dienstanforderungs-Eingabe-Connectors

Die Schrittfolge zum Erstellen eines Dienstanforderungs-Eingabe-Connectors entspricht der für andere Eingabe-Connectors in Design Center.

### Connector-Einstellungen




| Option | Beschreibung |
|---|---|
| **Anforderungstyp** | Wählen Sie die Option **Allgemein**. |
| **Dienstname** | Der Name des Webdienstes, der für LiveCycle ES2 offengelegt werden soll. |

## Aktivieren von Dienstantworten

In einem Szenario, in dem LiveCycle ES2 die endgültige Ausgabe bereitstellt, muss StreamServer so konfiguriert werden, dass es die Ausgabedaten an LiveCycle ES2 als Webdienstantwort zurückgibt. Dies wird in der Ausgabe-Connector-Konfiguration festgelegt. Es kann ein beliebiger Ausgabe-Connector-Typ verwendet werden, z. B. ein Null-Connector.

**Hinweis:** Der Ausgabe-Connector muss mit einer Ausgabewarteschlange verbunden sein.

### So aktivieren Sie Dienstantworten

1. Aktivieren Sie in Design Center die allgemeine Platform-Ebene.
2. Doppelklicken Sie auf den Ausgabe-Connector. Das Dialogfeld **Ausgabe-Connector-Einstellungen** wird geöffnet.
3. Klicken Sie auf das Symbol **Allgemein**, wählen Sie **Ergebnis in Dienstantwort einbeziehen**, und klicken Sie auf **OK**.

## Abrufen von Variablen

Wenn die Dienstantwort von LiveCycle ES2 Variablen enthält, muss StreamServer mithilfe der Skriptfunktion `GetConnectorValue` die Variablen abrufen. Weitere Informationen zu dieser Skriptfunktion finden Sie im StreamServe-Handbuch *Scripting reference*.

# Aufrufen von LiveCycle ES2-Prozessen aus LiveCycle Production Print ES2

StreamServer kann LiveCycle-Prozesse aufrufen, die in LiveCycle ES2 bereitgestellt und über Webdienste offengelegt werden. Mithilfe dieser Webdienste können bei der Dokumentverarbeitung LiveCycle-Prozesse in die StreamServer-Pipeline integriert werden.

Mithilfe von LiveCycle Workbench ES2 erstellte und aktivierte Prozesse können durch Senden der entsprechenden Aufrufanforderung (SOAP-Anforderung) an LiveCycle ES2 aufgerufen werden.

### LiveCycle-Filter und LiveCycle-Ausgabe-Connector

Es gibt zwei Möglichkeiten, Anforderungen von StreamServer an LiveCycle ES2 aufzurufen:

- LiveCycle-Ausgabe-Connector: Wird verwendet, wenn LiveCycle ES2 die endgültige Ausgabe bereitstellt. Informationen hierzu finden Sie unter *LiveCycle-Ausgabe-Connector* auf Seite 45.
- LiveCycle-Filter: Wird verwendet, wenn StreamServer die endgültige Ausgabe bereitstellt. Informationen hierzu finden Sie unter *LiveCycle-Filter* auf Seite 51.

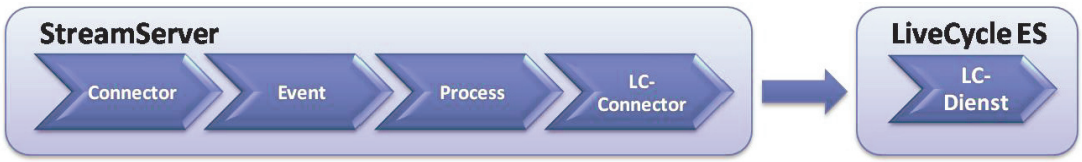
### Musterprojekt

Ein Design Center-Projekt mit dem Namen `sampleproject.dcpackage` steht auf dem Installationsmedium zur Verfügung. Das Projekt stellt diesen Einsatz musterhaft dar. Es veranschaulicht sowohl die Vorgänge auf der LiveCycle ES2-Prozessseite als auch auf der LiveCycle Production Print-Seite. Es ist ratsam, dieses Musterprojekt genau zu studieren und auszuprobieren. Das Musterprojekt ist auch hilfreich als Vorlage bei der Entwicklung eigener integrierter Prozesse. Weitere Informationen zum Musterprojekt finden Sie unter *Das Musterprojekt* auf Seite 59.

## LiveCycle-Ausgabe-Connector

Der LiveCycle-Ausgabe-Connector wird verwendet, wenn LiveCycle ES2 die endgültige Ausgabe bereitstellt.

### Beispiel: Verwendung des LiveCycle-Ausgabe-Connectors



1. StreamServer empfängt die Eingabedaten über einen Eingabe-Connector.
2. StreamServer verwendet die entsprechende Event/Process-Konfiguration zum Erstellen von Dokumenten.
3. Der LiveCycle-Ausgabe-Connector ruft den entsprechenden LiveCycle ES2-Process auf und sendet die angeforderten Dokumente.
4. Der LiveCycle ES2-Process verarbeitet die Dokumente und stellt die endgültige Ausgabe bereit.

## Erstellen eines Ausgabe-Connector-fähigen LiveCycle ES2-Process

Erstellen und aktivieren Sie den Process wie in der LiveCycle Workbench ES2-Dokumentation beschrieben. Um den LiveCycle-Ausgabe-Connector für das Aufrufen der bereitgestellten Dienste zu aktivieren, müssen Sie dem LiveCycle-Process die folgenden Eingabevariablen hinzufügen.

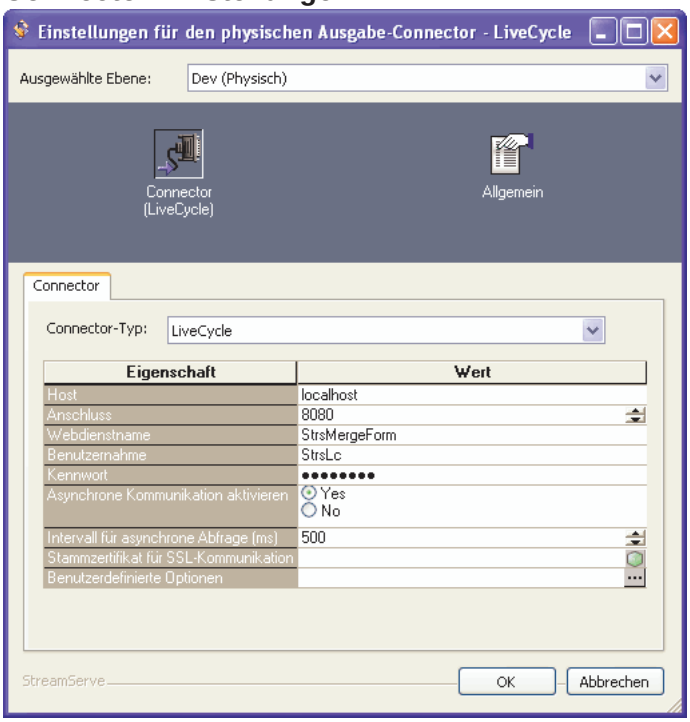
| Variablenname | Typ | Kommentar |
|---|---|---|
| `inputDoc` | `document` | Obligatorisch |
| `optionsMap` | `map` | Optional. Wird verwendet, wenn benutzerdefinierte Schlüssel in den LiveCycle-Filtereinstellungen angegeben werden. |

Nur entsprechend konfigurierte LiveCycle-Processes können von einem LiveCycle-Ausgabe-Connector aufgerufen werden.

## Erstellen eines LiveCycle-Ausgabe-Connectors

Einen LiveCycle-Ausgabe-Connector erstellen Sie auf dieselbe Weise wie andere Ausgabe-Connectors in Design Center.

### Connector-Einstellungen




| Option | Beschreibung |
|---|---|
| **Host** | Der Hostname oder die IP-Adresse des Servers, auf dem LiveCycle ES2 ausgeführt wird, z. B.:<br>`localhost` |
| **Anschluss** | Der vom LiveCycle ES2-Server verwendete Anschluss. Beispiel:<br>`8080` |
| **Webdienstname** | Der Name (Groß-/Kleinschreibung wird unterschieden) des aufzurufenden Dienstes. Der Name muss mit dem für den in LiveCycle Workbench ES2 erstellten zugehörigen Process identisch sein. |

| Option | Beschreibung |
|---|---|
| **Benutzername** | Benutzername, der für den Aufbau der Verbindung zu dem Server, auf dem LiveCycle ES2 ausgeführt wird, verwendet wird. Kommt bei einer einfachen HTTP-Authentifizierung zum Einsatz. |
| **Kennwort** | Kennwort, das für den Aufbau der Verbindung zu dem Server, auf dem LiveCycle ES2 ausgeführt wird, verwendet wird. Kommt bei einer einfachen HTTP-Authentifizierung zum Einsatz. |
| **Asynchrone Kommunikation aktivieren** | **Ja**<br>Ruft den Dienst asynchron auf. Diese Option wird verwendet, wenn langlebige LiveCycle-Dienste aufgerufen werden.<br>**Nein**<br>Ruft den Dienst synchron auf. Diese Option wird verwendet, wenn kurzlebige LiveCycle-Dienste aufgerufen werden. |
| **Asynchrones Abfrageintervall** | Wird nur zusammen mit asynchronen Aufrufen verwendet. Dies ist das Intervall (in Millisekunden), in dem überprüft wird, ob eine Antwort auf eine Aufrufanforderung vorliegt. |
| **Stammzertifikat für SSL-Kommunikation** | Das bei Nutzung von HTTPS als Webdienstprotokoll verwendete Stammzertifikat (sichere Kommunikation). Das Zertifikat muss über ein mit der Platform verbundenes Ressourcenset verfügbar sein. |

| Option | Beschreibung |
|---|---|
| **Benutzerdefinierte Optionen** | Eine Liste benutzerdefinierter Schlüssel (Schlüssel/Wert-Paare), die in der Aufrufanforderung verwendet werden können.<br><br>Um benutzerdefinierte Schlüssel verarbeiten zu können, muss der Dienst über eine Variable `optionsMap` vom Typ `map` verfügen. Alle hier festgelegten benutzerdefinierten Schlüssel werden der Variablen `optionsMap` im aufgerufenen Dienst hinzugefügt.<br><br>Die bereitgestellten Werte können im eingehenden LiveCycle-Process mithilfe eines XPath-Ausdrucks im LiveCycle-Process extrahiert werden.<br><br>Beispiele für benutzerdefinierte Schlüssel sind Kennwörter zum Erstellen kennwortverschlüsselter PDF-Dateien. Beispiel:<br><br>**Schlüssel:** `pdfpassword`<br><br>**Wert:** `encrypted` |

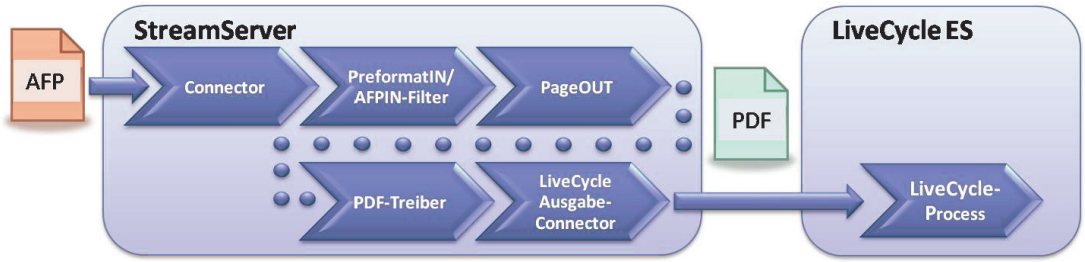
# Verwendungsszenario

### Hintergrund

Ein Geschäftsprozess verlangt, dass eine AFP-Datei mit Rechnungen in eine PDF-Datei konvertiert und anschließend an LiveCycle ES2 zur Speicherung in LiveCycle Content Services ES2 weitergegeben wird.

### Aktionen

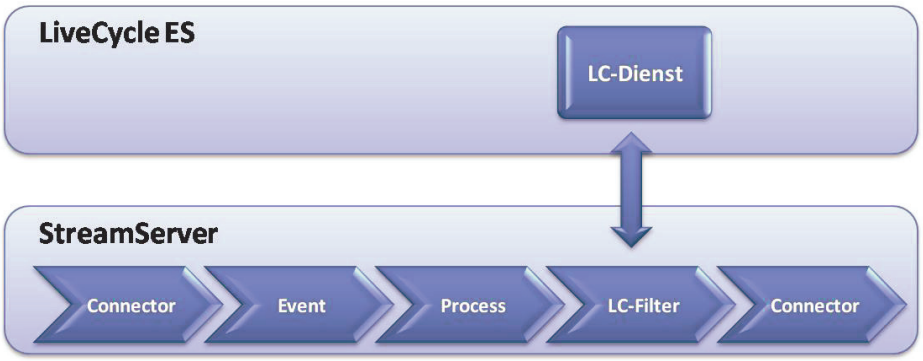
StreamServer wird der Pipeline hinzugefügt. StreamServer ruft die AFP-Eingabedaten über einen Eingabe-Connector, einen AFPIN-Filter und ein PreformatIN-Event ab. Die AFP-Daten werden dann über einen PageOUT-Process und einen PDF-Treiber in PDF-Daten umgewandelt. Die PDF-Ausgabe wird abschließend über einen LiveCycle-Ausgabe-Connector an einen LiveCycle-Process übergeben.



# LiveCycle-Filter

Der LiveCycle-Filter wird verwendet, wenn StreamServer die endgültige Ausgabe bereitstellt.

### Beispiel: Verwendung des LiveCycle-Filters




1. StreamServer empfängt die Eingabedaten über einen Eingabe-Connector.
2. StreamServer verwendet die entsprechende Event/Process-Konfiguration zum Erstellen von Dokumenten.
3. Der LiveCycle-Filter ruft den entsprechenden LiveCycle ES2-Process auf und sendet die angeforderten Dokumente.
4. Der LiveCycle ES2-Process verarbeitet die Dokumente und sendet die verarbeiteten Dokumente als Antwort an StreamServer.
5. StreamServer stellt die endgültige Ausgabe über einen Ausgabe-Connector bereit.

Wenn der Webdienst unterbrochen wird, bevor die Dokumente in der Webdienstantwort gesendet wurden, wird keine Ausgabe bereitgestellt. In diesem Fall wird eine Fehlermeldung protokolliert.

## Erstellen eines filterfähigen LiveCycle ES2-Process

Erstellen und aktivieren Sie den Process wie in der LiveCycle Workbench ES2-Dokumentation beschrieben. Um den LiveCycle-Filter für das Aufrufen der bereitgestellten Dienste zu aktivieren, müssen Sie dem LiveCycle-Process die folgenden Eingabe- und Ausgabevariablen hinzufügen.

| Eingabe/ Ausgabe | Variablenname | Typ | Kommentar |
|---|---|---|---|
| Eingabe | `inputDoc` | `document` | Obligatorisch |
| Eingabe | `optionsMap` | `map` | Optional. Wird verwendet, wenn benutzerdefinierte Schlüssel in den LiveCycle-Filtereinstellungen angegeben werden. |
| Ausgabe | `outputDoc` | `document` | Obligatorisch |

Nur entsprechend konfigurierte LiveCycle-Processes können von einem LiveCycle-Filter aufgerufen werden.

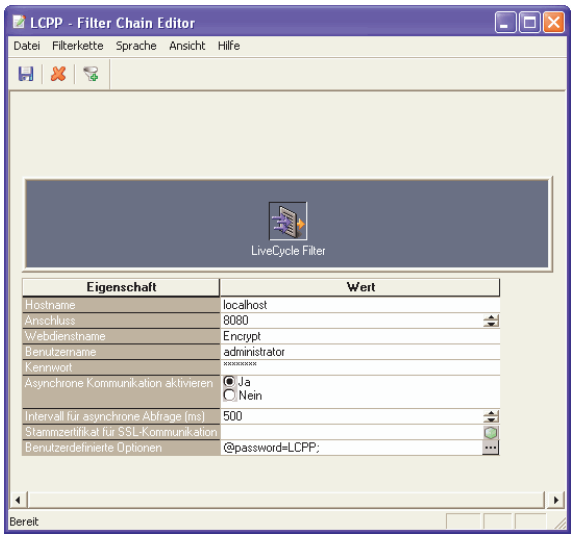
## Erstellen eines LiveCycle-Filters

Einen LiveCycle-Filter erstellen Sie auf dieselbe Weise wie andere Ausgabefilter in Design Center. Das bedeutet, Sie müssen eine Filterkettenressource erstellen, den LiveCycle-Filter in der Filterkette erstellen und konfigurieren und die Filterkette mit dem geeigneten Ausgabe-Connector verbinden.

### So erstellen Sie einen LiveCycle-Filter und wenden ihn an

1. Erstellen Sie in einem mit der Platform verbundenen Ressourcenset eine Filterkettenressource.
2. Fügen Sie der Filterkette einen LiveCycle-Filter hinzu.
3. Konfigurieren Sie den Filter (siehe „Filtereinstellungen“ weiter unten), und speichern Sie die Filterkettenressource.
4. Fügen Sie die Filterkette dem geeigneten Ausgabe-Connector hinzu.

### Filtereinstellungen




| Option | Beschreibung |
|---|---|
| **Hostname** | Der Hostname oder die IP-Adresse des Servers, auf dem LiveCycle ES2 ausgeführt wird, z. B.:<br>`localhost` |
| **Anschluss** | Der vom LiveCycle ES2-Server verwendete Anschluss. Beispiel:<br>`8080` |

| Option | Beschreibung |
|---|---|
| **Webdienstname** | Der Name (Groß-/Kleinschreibung wird unterschieden) des aufzurufenden Dienstes. Der Name muss mit dem für den in LiveCycle Workbench ES2 erstellten zugehörigen Process identisch sein. |
| **Benutzername** | Benutzername, der für den Aufbau der Verbindung zu dem Server, auf dem LiveCycle ES2 ausgeführt wird, verwendet wird. Kommt bei einer einfachen HTTP-Authentifizierung zum Einsatz. |
| **Kennwort** | Kennwort, das für den Aufbau der Verbindung zu dem Server, auf dem LiveCycle ES2 ausgeführt wird, verwendet wird. Kommt bei einer einfachen HTTP-Authentifizierung zum Einsatz. |
| **Asynchrone Kommunikation aktivieren** | **Ja**<br>Ruft den Dienst asynchron auf. Diese Option wird verwendet, wenn langlebige LiveCycle-Dienste aufgerufen werden.<br>**Nein**<br>Ruft den Dienst synchron auf. Diese Option wird verwendet, wenn kurzlebige LiveCycle-Dienste aufgerufen werden. |
| **Asynchrones Abfrageintervall** | Wird nur zusammen mit asynchronen Aufrufen verwendet. Dies ist das Intervall (in Millisekunden), in dem überprüft wird, ob eine Antwort auf eine Aufrufanforderung vorliegt. |
| **Stammzertifikat für SSL-Kommunikation** | Das bei Nutzung von HTTPS als Webdienstprotokoll verwendete Stammzertifikat (sichere Kommunikation). Das Zertifikat muss über ein mit der Platform verbundenes Ressourcenset verfügbar sein. |

| Option | Beschreibung |
|---|---|
| **Benutzerdefinierte Optionen** | Eine Liste benutzerdefinierter Schlüssel (Schlüssel/Wert-Paare), die in der Aufrufanforderung verwendet werden können.<br><br>Um benutzerdefinierte Schlüssel verarbeiten zu können, muss der Dienst über eine Variable `optionsMap` vom Typ `map` verfügen. Alle hier festgelegten benutzerdefinierten Schlüssel werden der Variablen `optionsMap` im aufgerufenen Dienst hinzugefügt.<br><br>Die bereitgestellten Werte können im eingehenden LiveCycle-Process mithilfe eines XPath-Ausdrucks im LiveCycle-Process extrahiert werden.<br><br>Beispiele für benutzerdefinierte Schlüssel sind Kennwörter zum Erstellen kennwortverschlüsselter PDF-Dateien. Beispiel:<br><br>**Schlüssel:** `pdfpassword`<br><br>**Wert:** `encrypted` |

## Verwendungsszenario

### Hintergrund

Ein StreamServer-Benutzer muss PDF-Dokumente verschlüsseln. Die Verschlüsselung kann jedoch nicht mithilfe von StreamServer-Funktionen erfolgen. Diese Funktionalität kann durch das Aufrufen eines LiveCycle-Process zum Verschlüsseln des PDF-Dokuments erreicht werden.

*Abbildung 2 Vor-LiveCycle-Filter*

### Aktionen

Ein LiveCycle-Filter wird hinter dem PDF-Treiber eingefügt. Der LiveCycle-Filter sendet die formatierten PDF-Dokumente an einen LiveCycle-Process. Der LiveCycle-Process verschlüsselt die Dokumente und gibt die verschlüsselten Dokumente über den LiveCycle-Filter an StreamServer zurück.

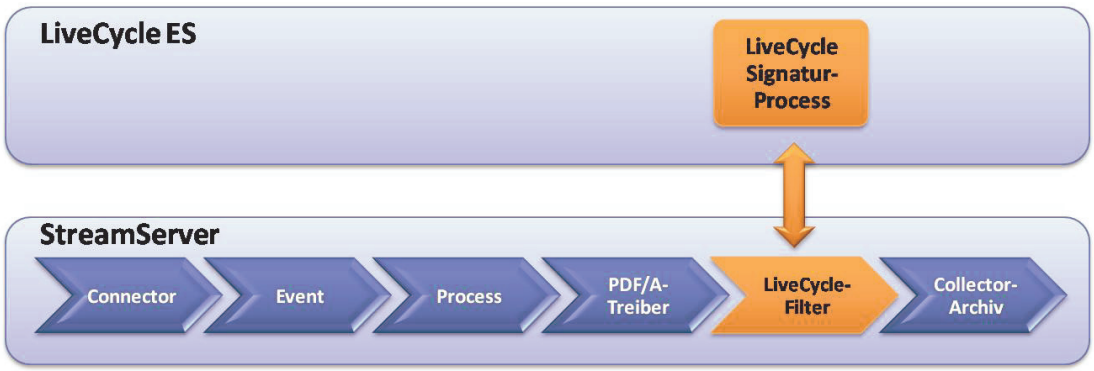



*Abbildung 3 Nach-LiveCycle-Filter*

# Globale LiveCycle-Filter- und LiveCycle-Connector-Einstellungen

Neben den in der Benutzeroberfläche für den LiveCycle-Filter und den LiveCycle-Ausgabe-Connector konfigurierten Einstellungen müssen Sie möglicherweise auch einige globale Einstellungen in der Konfigurationsdatei `strslcfilter.config.xml` ändern. Diese Konfigurationsdatei finden Sie unter:

`<StreamServe-Installationsverzeichnis>\Services\XFA\1.3.0\Service`

```
<?xml version="1.0" encoding="utf-8"?>
<lcfilter>
  <setting key="maxinlinesize">65536</setting>
  <setting key="timeout">120</setting>
  <setting key="retries">5</setting>
 </lcfilter>
```

*Abbildung 4 Beispiel für die Datei „strslcfilter.config.xml“*

Diese Einstellungen gelten für alle LiveCycle-Filter und LiveCycle-Ausgabe-Connectors.

| Schlüssel | Beschreibung |
|---|---|
| `maxinlinesize` | Die maximale für ein Dokument zulässige Größe (in Byte) für die base64-Inline-Codierung. Bei Überschreiten dieser Größe wird das Dokument stattdessen als DIME-Anhang gespeichert. |
| `timeout` | Die Zeitdauer (in Sekunden), die auf eine Antwort auf die Anforderung gewartet wird. Bei Überschreiten dieser Zeitdauer wird die Verbindung geschlossen. |
| `retries` | Die Anzahl der Wiederholversuche im Fall von Kommunikationsfehlern. |

# Das Musterprojekt

Das Musterprojekt ist zur Verwendung für folgende Zwecke konfiguriert:

- zu Schulungszwecken.
- zum Überprüfen der ordnungsgemäßen Installation und Konfiguration von LiveCycle Production Print ES2.

Das Musterprojekt zeigt auch, wie LiveCycle ES2- und LiveCycle Production Print ES2-Processes integriert werden können. Die bereitgestellten Beispielintegrationen können als Muster für Ihre Implementierung verwendet werden.

Das Musterprojekt `SampleProject.dcpackage` steht auf dem Installationsmedium im Ordner `Extras\sampleproject` zur Verfügung. Das Musterprojekt wird standardmäßig auch im Verzeichnis `C:<StreamServe-Installationsverzeichnis>\Services\XFA\1.3.0\Tool` installiert.

### Voraussetzungen

Zum Ausführen des Musterprojekts müssen die Processes und Ressourcen des Beispielprojekts installiert sein.

Wenn Sie einen LiveCycle-Server ausführen (d. h. StreamServer-Anwendungen aus LiveCycle ES2 aufrufen), muss die DSC installiert sein.

Informationen hierzu finden Sie im *LiveCycle Production Print ES2-Installationshandbuch*.

### In diesem Kapitel behandelte Themen


- *Ausführen des Musterprojekts* auf Seite 60
- *Überprüfen der Basisinstallation* auf Seite 61
- *Musterprojektkonfigurationen* auf Seite 62

# Ausführen des Musterprojekts

Um das Musterprojekt exportieren und bereitstellen zu können, müssen Sie es zunächst entpacken.

### So entpacken Sie die Musterprojektdatei

1. Öffnen Sie Design Center.
2. Wählen Sie **Datei > Projekt entpacken** aus.
3. Suchen und öffnen Sie die Paketdatei. Das Dialogfeld **Projekt entpacken** wird geöffnet.
4. Geben Sie an, wo die Projektdateien entpackt werden sollen, und klicken Sie auf **OK**.

### So exportieren Sie das Musterprojekt und stellen es bereit

Exportieren Sie das Projekt in Design Center. Weitere Informationen finden Sie in der Dokumentation zu *StreamServe Design Center*.

### So stellen Sie das Musterprojekt bereit

In Control Center:

1. Erstellen Sie die StreamServer-Anwendung, für die das Musterprojekt bereitgestellt werden soll.
2. Stellen Sie das Musterprojekt für die StreamServer-Anwendung bereit.

Weitere Informationen finden Sie in der Dokumentation zu *StreamServe Control Center*.

### So starten Sie die StreamServer-Anwendung

Starten Sie die StreamServer-Anwendung in Control Center, indem Sie mit der rechten Maustaste auf den StreamServer-Anwendungsknoten klicken und die Option **Start** wählen.

Das Stoppen und erneute Bereitstellen von StreamServer-Anwendungen erfolgt ebenfalls aus Control Center heraus. Weitere Informationen finden Sie in der Dokumentation zu *StreamServe Control Center*.

# Überprüfen der Basisinstallation

### So überprüfen Sie die Grundfunktionen der Installation

1. Entpacken, exportieren und stellen Sie das Projekt bereit. Folgen Sie den Anweisungen in *Ausführen des Musterprojekts* auf Seite 60.
2. Erstellen Sie den Ordner `input` im Verzeichnis `C:\ManagementGateway\1.0\root\applications\<`*`Name Ihrer Anwendung`*`>\Dev`
3. Kopieren Sie die Datei `Purchase Order.xml` aus dem Verzeichnis `C:\ManagementGateway\1.0\root\applications\<`*`Name Ihrer Anwendung`*`>\data\samples` in den Ordner `input`.
4. Vergewissern Sie sich anhand des Protokolls in Control Center, dass die Datei verarbeitet wurde.
5. Vergewissern Sie sich, dass die Datei `purchaseorder.pdf` im Verzeichnis `C:\ManagementGateway\1.0\root\applications\<`*`Name Ihrer Anwendung`*`>\Dev\output` vorhanden ist.

### So überprüfen Sie das Musterprojekt für die AFP-nach-PDF-Konvertierung

1. Erstellen Sie einen Ordner mit der Bezeichnung `input2` im Verzeichnis `C:\ManagementGateway\1.0\root\applications\<`*`Name Ihrer Anwendung`*`>\Dev`
2. Kopieren Sie die Datei `purchaseorder.afp` aus dem Verzeichnis `C:\ManagementGateway\1.0\root\applications\<`*`Name Ihrer Anwendung`*`>\data\samples` in den Ordner `input2`.
3. Vergewissern Sie sich anhand des Protokolls in Control Center, dass die Datei verarbeitet wurde.
4. Vergewissern Sie sich, dass sich fünf Dateien mit der Bezeichnung `nnnnnnnnn.pdf` im Verzeichnis `C:\ManagementGateway\1.0\root\applications\<`*`Name Ihrer Anwendung`*`>\Dev\output` befinden, wobei `nnnnnnn` für die Bestellnummer aus den Eingabedaten steht.

# Musterprojektkonfigurationen

### Musterkonfiguration 1

Die Musterkonfiguration 1 zeigt einen grundlegenden Anwendungsfall mit synchroner Nachbearbeitung, einschließlich Dokumentsortierung und OMR-Markierung.

Sie können den Process mit dem **PDF encrypt**-Ausgabe-Connector verbinden, das LCA-Musterpaket installieren und den Filter für den Connector so konfigurieren, dass er auf den Computer zeigt, auf dem LCA installiert ist. Dieses Muster zeigt, wie LiveCycle Production Print ES2 eine Verbindung mit einem LiveCycle-Server herstellen, eine PDF-Datei verarbeiten und diese PDF-Datei anschließend an LiveCycle Production Print ES2 zurücksenden kann.

### Musterkonfiguration 2

Die Musterkonfiguration 2 zeigt, wie ASCII-Daten an eine Formularvorlage gebunden werden.

1. Kopieren Sie die Datei `Invoice.grb` aus dem Musterverzeichnis in das Eingabeverzeichnis.
2. Überprüfen Sie, ob die Datei `output.pdf` im Ausgabeverzeichnis erstellt wurde.

### Process-Konfiguration für Mahnbescheide

Die Musterkonfiguration 1 und die Process-Konfiguration für Mahnbescheide können in Kombination verwendet werden, um den Remoteaufruf von LiveCycle Production Print ES2 aus LiveCycle ES2 darzustellen.

1. Installieren Sie das LCA-Paket auf einem LiveCycle ES2-Server, und stellen Sie es bereit.
2. Rufen Sie den Dienst `DunningNoticeStage` mit der Datei `DunningNotice.xml` als Eingabe (verfügbar in Ressourcen) auf. Die XML-Datei und die Datei `DunningNotice.xdp` werden an LiveCycle Production Print ES2 gesendet.

   Die Datendatei und die Vorlage werden zusammengeführt und im Post-Processor-Repository von LiveCycle Production Print ES2 abgelegt.
3. Rufen Sie den Dienst *`DunningNoticePrint`* mit der Datei `lcpp.ppq` (Post-Processor-Abfragedatei) als Eingabe auf. Die PPQ-Datei wird an LiveCycle Production Print ES2 gesendet, das wiederum Mahnbescheide der jeweiligen Mahnstufe aus dem Post-Processor-Repository abruft, diese als PDF-Datei formatiert und in den Ausgabeordner als `output.pdf` schreibt.

### Generierungskonfiguration für Mahnbescheide

Aus der Generierungskonfiguration für Mahnbescheide wird deutlich, wie LiveCycle Production Print ES2 mithilfe von Daten und einer Vorlage aufgerufen werden kann.

LiveCycle Production Print ES2 führt die Daten und die Vorlage zusammen und gibt das Ergebnis als AFP-Datei an LiveCycle ES2 zurück.

1 Rufen Sie den Dienst `DunningNoticeGenerate` mit der Datei `Dunning Notice.xml` als Eingabe auf. Die resultierende AFP-Datei wird in eine Datei auf dem LiveCycle-Server geschrieben (standardmäßig `c:\result.afp`).

### Musterkonfiguration 5: AFP2PDF

Aus der Musterkonfiguration 5 geht hervor, wie AFP-Dateien mit dem PreformatIN-Tool und dynamischen Overlays im PageOUT-Process in das PDF-Format konvertiert werden können.

Das PreformatIN-Event liest die Datei `PurchaseOrder.afp` durch einen AFP2LXF-Filter und versucht über einen Musterabgleich, den Dokumenttyp zu ermitteln. Der PageOUT-Process ordnet dann die dynamisch generierten LXF-Seiten jeweils logischen Seiten zu. Das Ergebnis wird über einen Treiber im Ausgabe-Connector als PDF gerendert.

1 Rufen Sie die AFP2PDF-Konfiguration auf, indem Sie die Datei `purchaseorder.afp` an das Eingabeverzeichnis `Watchfolder2` senden. Im Ergebnis dieses Vorgangs werden fünf PDF-Dateien in das Ausgabeverzeichnis ausgegeben. Dabei entspricht jede PDF jeweils einem Dokument in der AFP-Datei.

## Die Platform-Konfiguration

Die Platform im Musterprojekt enthält Eingabe- und Ausgabe-Connectors in der physischen Ebene mit dem Namen `Dev`.

### Eingabe-Connectors

| | |
|---|---|
| `input` | Connector zum Scannen von Verzeichnissen. Bei dem gescannten Ordner handelt es sich um `.\input` (relativ zum Arbeitsverzeichnis eines bereitgestellten Projekts). |
| `DunningNoticeStage`<br>`DunningNoticePrint`<br>`DunningNoticeGenerate` | Dienstanforderungs-Connectors. Legt Message-Konfigurationen als Dienste über das StreamServe Services Gateway offen. |
| `WatchFolder2` | Connector zum Scannen von Verzeichnissen.<br>Das gescannte Verzeichnis ist `.\input2` (bezogen auf das Arbeitsverzeichnis). Gescannt wird auf den Dateityp `*.afp`. Der AFP2LXF-Filter wird auf die Eingabe-Pipeline des Connectors angewendet. |

### So zeigen Sie die Connector-Einstellungen an

1. Aktivieren Sie die Platform-Ansicht.
2. Klicken Sie mit der rechten Maustaste auf den Connector, und wählen Sie die Option **Einstellungen** aus.
3. Wechseln Sie im Dialogfeld **Einstellungen** zwischen der logischen und der physischen Ebene, um alle Connector-Einstellungen anzuzeigen.

### Ausgabe-Connectors

| | |
|---|---|
| `PDF` | Datei-Connector.<br>Als Pfad der Ausgabedatei ist `.\output\purchase-order.pdf` angegeben (jeweils im Arbeitsverzeichnis eines bereitgestellten Projekts). Der Gerätetreiber ist auf den Connector `PDF` mit Standardoptionen festgelegt.<br>Für den Ausgabemodus ist `Job` eingestellt, damit alle Eingabedaten in einer einzigen Ausgabedatei gespeichert werden. |
| `PDF encrypt` | Datei-Connector mit den gleichen Einstellungen wie der `PDF`-Connector. Der Unterschied besteht darin, dass der `PDF encrypt`-Connector über eine Filterkonfiguration verfügt, die einen LiveCycle-Dienst per Remotezugriff aufrufen kann. Damit ist es möglich, die Ausgabedaten zu verschlüsseln und durch ein Kennwort zu schützen, bevor sie in die Datei geschrieben werden. |
| `PostProcessing` | Datei-Connector.<br>Als Pfad der Ausgabedatei ist `.\output\output.pdf` angegeben (relativ zum Arbeitsverzeichnis eines bereitgestellten Projekts). Der Gerätetreiber ist auf den Connector `PDF` mit Standardoptionen festgelegt.<br>Für den Ausgabemodus ist `Job` eingestellt, damit alle Eingabedaten in einer einzigen Ausgabedatei gespeichert werden. |
| `PPRepository` | Post-Processor-Repository-Connector.<br>Speichert Ausgaben im eingebetteten Post-Processor-Repository mithilfe des Alias `lcpp`. Der Gerätetreiber ist auf `SDR` eingestellt. |
| `LC Response` | Ein Null-Connector, für den die Option **Ergebnis in Dienstantwort einbeziehen** aktiviert ist. Der Gerätetreiber ist auf `AFP` eingestellt. |
| `PDF-Bypass` | Vergleichbar mit dem PDF-Connector. Der Unterschied besteht darin, dass der Ausgabemodus auf „Process“ gesetzt wird. |

# Das Ressourcenset

Das Standardressourcenset im Musterprojekt enthält für jeden der drei Message-Typen jeweils einen Ordner:

- `Invoice`
- `Purchase Order`
- `Dunning Notice`

Es sind Musterressourcen für die Konfiguration der Messages vorhanden:

- `/Invoice/invoice.grb` ist eine Textdruckdatei, die als Eingabe für die Message-Musterkonfiguration 2 (ASCII-Eingabedaten) dient
- `/Invoice/invoice.xdp` ist die von der Message-Musterkonfiguration 2 für ASCII-Eingabedaten verwendete Vorlage.
- `/Purchase Order/Purchase Order.xml` dient als Eingabe für die Message-Musterkonfiguration 1 (XML-Eingabedaten).
- `/Purchase Order/Purchase Order.xdp` ist eine vordefinierte Formularvorlage zur Verwendung mit der Message-Musterkonfiguration 1 für XML-Eingabedaten.
- `/Dunning Notice/Dunning notice.xml` ist eine Eingabe für die Message-Musterkonfigurationen 3 (Aufruf durch LiveCycle) und 4 (Generieren eines Mahnbescheids).
- `/AFP2PDF/AFP2PDF`, Filter-Pipeline mit konfiguriertem AFP2LXF-Filter.
- `/AFP2PDF/purchaseorder.afp` Mustereingabedatei im AFP-Format für die Musterkonfiguration 5 (AFP2PDF).

# Die Messages

### Muster 1 – XML-Eingabedaten (mithilfe einer vorhandenen Formularvorlage)

Bei dieser Message kommt eine statische Vorlage (`purchase order.xdp`) zum Einsatz. Die Formularvorlage verwendet als Eingabe eine XML-Datendatei.

Für die Message wurden ein Event und ein Process konfiguriert. Das Event ermittelt den Datendateityp für die Message mithilfe eines Musters. Das Muster `/batch` gewährleistet die Übereinstimmung mit dem Stammknoten der Datendatei `Purchase Order.xml`. Wenn die Eingabedaten mit dem Muster des Events übereinstimmen, wird der Process mit einer Formularvorlage verknüpft. Die Formularvorlage `Purchase Order.xdp` aus dem Ressourcenset wird in den Process geladen.

Einstellungen:

- Der Datensatzmodus wird mit dem Datensatz-Trigger `transaction` verwendet.
- Mithilfe der StreamServe-Variablenzuordnung wird der SOM-Ausdruck `$record.header.txtPONum` der StreamServe-Variable `$ponum` zugeordnet. Die Variable wird in den Runtime-Einstellungen zum Sortieren der Formulare in der Batch-Eingabedatei verwendet.

### Muster 2 – ASCII-Eingabedaten (mithilfe einer in Design Center manuell erstellten Vorlage)

Die Eingabedaten für die Formularvorlage befinden sich in einer Textdatei. Das Event verwendet das PageIN-Tool zum Erkennen des Datendateityps und zum Extrahieren des Inhalts der Datendatei in eine StreamServe-Message.

Zur Konfiguration des Events dient die Musterdatei `invoice.grb` aus dem Ressourcenset. Anhand eines Musters wird die Zeichenfolge `INVOICE` in einer Reihe von Koordinaten auf der Seite erkannt. Das Feld-Tool in PageIN wurde eingesetzt, um Daten aus den Koordinaten auf der Seite in die Message zu extrahieren.

Der Process verwendet die Einstellung **Message als Datenverbindung in der Datenansicht hinzufügen** (siehe *Dialogfeld „Einstellungen“* auf Seite 86), um den Inhalt der Message in der Datenansicht von LiveCycle Designer anzuzeigen. Mithilfe des Inhalts der Datenansicht wurden Bindungen zu den Feldern in der Formularvorlage erstellt.

### Muster 3 – Aufruf durch LiveCycle (Remoteaufruf von LiveCycle Production Print ES2)

Als Eingabedaten für das XMLIN-Event wird der Inhalt der XML-Datei `Dunning Notice.xml` verwendet. Das Muster ist auf die Erkennung des Elements `<transaction>` ausgelegt. Der Process ist für das dynamische Laden einer Vorlage konfiguriert. Der Variablen `$template` wird aus dem Eingabe-Connector automatisch eine Vorlage zugewiesen.

### Muster 4 – Generieren eines Mahnbescheids (Remoteaufruf von LiveCycle Production Print ES2)

Diese Message-Konfiguration ist mit Muster 3 identisch. Der Unterschied zwischen beiden Konfigurationen besteht lediglich in deren Verwendung in der Runtime.

### Muster 5 – AFP2PDF

In dieser Message-Konfiguration werden AFP-Dateien mithilfe von PreformatIN und PageOUT und dem AFP2LXF-Filter der Platform in PDF-Dateien konvertiert.

## Die Runtime-Konfigurationen

### Job-Konfiguration für Bestellungen (Purchase Order)

Für diese Job-Konfiguration wird das Message-Muster 1 verwendet.
Das Event wird mit dem Eingabe-Connector `WatchFolder` verbunden.

Die Auswahlmethode für den Ausgabe-Connector ist auf `Statisch` gesetzt, und es wird der Ausgabe-Connector `PostProcessing` verwendet. Für den Ausgabe-Connector `PostProcessing` sind folgende Einstellungen festgelegt:

- Die OMR-Markierung auf der Registerkarte für den Process-Beginn.

- Die Dokumentsortierung basiert auf der Variablen `$ponum`. Es ist die Sortierung in absteigender Reihenfolge festgelegt.

### Job-Konfiguration für Rechnungen (Invoice)

Für diesen Job wird das Message-Muster 2 verwendet. Das Event wird mit dem Eingabe-Connector `WatchFolder` verbunden.

Die Auswahlmethode für den Ausgabe-Connector ist auf `Statisch` gesetzt, und es wird der Ausgabe-Connector `PDF` verwendet.

### Job-Konfiguration für Mahnstufen (Stage)

Für diesen Job wird das Message-Muster 3 verwendet. Das Event wird mit dem Eingabe-Connector `DunningNoticeStage` verbunden.

Die Auswahlmethode für den Ausgabe-Connector ist auf `Statisch` gesetzt, und es wird der Ausgabe-Connector `PPRepository` verwendet.

### Job-Konfiguration zum Drucken von Mahnbescheiden (Print)

Dies ist eine Post-Processor-Repository-Konfiguration. Der Job wird mit dem Eingabe-Connector `DunningNoticePrint` verbunden.

Die Auswahlmethode für den Ausgabe-Connector ist auf `Statisch` gesetzt, und es wird der Ausgabe-Connector `PDF` verwendet.

### Job-Konfiguration zum Generieren von Mahnbescheiden (Generate)

Für diesen Job wird das Message-Muster 4 verwendet. Das Event wird mit dem Eingabe-Connector `DunningNoticeGenerate` verbunden.

Die Auswahlmethode für den Ausgabe-Connector ist auf `Statisch` gesetzt, und es wird der Ausgabe-Connector `LC Response` verwendet.

### Job AFP2LXF

Durch diesen Job werden die Runtime-Connector-Einstellungen des PDF-Bypass-Connectors so konfiguriert, dass den PDF-Dateien über eine Variable eindeutige Dateinamen zugeordnet werden.

# Unterstützung für Etikettendrucker

LiveCycle Production Print unterstützt die folgenden Etikettendrucker:

- ZPL II – Ausführliche Informationen finden Sie unter *ZPL II* auf Seite 70.
- Intermec FP/DP – Ausführliche Informationen finden Sie unter *Intermec FP/DP* auf Seite 72.
- Intermec IPL – Ausführliche Informationen finden Sie unter *Intermec IPL* auf Seite 75.
- Printronix PGL/IGP – Ausführliche Informationen finden Sie unter *Printronix PGL/IGP* auf Seite 78.
- TEC – Ausführliche Informationen finden Sie unter *TEC* auf Seite 80.

### Einschränkungen bei der Unterstützung von Etikettendruckern

Es bestehen Einschränkungen für Etikettendruckersprachen bei der Unterstützung von Objekten und Objekteigenschaften im Vergleich zu Formaten wie PDF, PCL, AFP, PS usw.

Aus Leistungsgründen bietet LiveCycle Production Print nicht die gleiche WYSIWYG-Unterstützung wie die Vorschau in Designer ES2. Beispiel:

- Gradientenfüllungen werden nicht gerastert.
- Linienstile, z. B. gestrichelt, werden nicht für alle Druckersprachen unterstützt.

# ZPL II

## Textausgabe – ZPL II

### Skalierbare Schriftarten

Die folgenden skalierbaren Schriftarten werden unterstützt:

- CG Triumvirate Bold Condensed

Dies ist die einzige Schriftart, die dem Benutzer WYSIWYG bietet.

### Adobe-Schriftarten

Alle Adobe-Schriftarten werden der oben genannten Schriftart zugeordnet.

### ASCII

Es wird nur die US-amerikanische ASCII-Tabelle unterstützt.

## Barcodeunterstützung – ZPL II

Folgende Barcodes werden unterstützt:

- Aztec
- Codabar
- Code 11
- Code 128
- Code 2 of 5 Industrial
- Code 2 of 5 Interleaved
- Code 2 of 5 Standard
- Code 3 of 9
- Code 49
- Code 93
- DataMatrix
- EAN13
- EAN8
- Logmars
- MSI
- PDF417
- Planet Code
- Plessey
- QR Code
- RSS14
- RSS14 Expanded
- RSS14 Limited
- RSS14 Stacked
- RSS14 Stacked Omnidirectional
- RSS14 Truncated
- UPC-A
- UPC-E
- UPS Maxicode
- US Postal DPBC
- US Postal Standard
- US Postal Zip-5

# RFID-Unterstützung – ZPL II

RFID-Barcode wird unterstützt.

| | |
|---|---|
| **Funkprotokoll** | EPC Class 1 Generation 2. |
| **Barcodewert** | Behandelt als HEX96. |

# Unterstützung von Benutzeroberflächenobjekten – ZPL II

Unterstützte Benutzeroberflächenobjekte sind nachfolgend aufgelistet.

| **Objekt** | **Kommentar** |
|---|---|
| **Vertikale Linien** | Unterstützt. |
| **Horizontale Linien** | Unterstützt. |
| **Diagonale Linien** | Unterstützt. |
| **Rechtecke** | Unterstützt. |
| **Kreise** | Unterstützt. |
| **Bilder** | Unterstützt. |
| **Linienstil** | Es werden nur durchgezogene Linien unterstützt. |
| **Drehung** | Es werden Drehungen um 90°, 180° und 270° unterstützt. |

# Intermec FP/DP

## Textausgabe – Intermec FP/DP

### Skalierbare Schriftarten

Die folgenden skalierbaren Schriftarten werden unterstützt:

- Century Schoolbook BT
- Dutch 801 Bold BT
- Dutch 801 Roman BT
- Futura Light BT
- Letter Gothic 12 Pitch BT
- Monospace 821 Bold BT
- Monospace 821 BT
- OCR-A BT
- OCR-B 10 Pitch BT
- Prestige 12 Pitch Bold BT
- Swiss 721 Bold BT
- Swiss 721 BT
- Swiss 721 Condensed BT
- Zurich Extra Condensed Bold

Dies sind die einzigen Schriftarten, die dem Benutzer WYSIWYG bieten.

### Adobe-Schriftarten

Alle Adobe-Schriftarten werden den oben genannten Schriftarten zugeordnet (Schriftarten mit ähnlichen Metriken).

### ASCII

Es wird nur die US-amerikanische ASCII-Tabelle unterstützt.

## Barcodeunterstützung – Intermec FP/DP

Folgende Barcodes werden unterstützt:

- Aztec
- Codabar
- Code 11
- Code 128
- Code 2 of 5 Industrial
- Code 2 of 5 Interleaved
- Code 2 of 5 Matrix
- Code 2 of 5 Standard
- Code 3 of 9, Code 93
- Code 49, MSI, Planet Code
- DataMatrix
- EAN13
- EAN8
- PDF417
- Plessey
- QR Code
- RSS14
- RSS14 Expanded
- RSS14 Limited
- RSS14 Stacked
- RSS14 Stacked Omnidirectional
- RSS14 Truncated
- UPC-A
- UPC-E, UPS Maxicode
- US Postal DPBC
- US Postal Standard
- US Postal Zip-5

## RFID-Unterstützung – Intermec FP/DP

RFID-Barcode wird unterstützt.

| | |
|---|---|
| **Funkprotokoll** | EPC Class 1 Generation 2. |
| **Barcodewert** | Behandelt als HEX96. |

# Unterstützung für Benutzeroberflächenobjekte – Intermec FP/DP

Unterstützte Benutzeroberflächenobjekte sind nachfolgend aufgelistet.

| Objekt | Kommentar |
|---|---|
| **Vertikale Linien** | Unterstützt. |
| **Horizontale Linien** | Unterstützt. |
| **Diagonale Linien** | Nicht unterstützt. |
| **Rechtecke** | Unterstützt. |
| **Kreise** | Nicht unterstützt. |
| **Bilder** | Unterstützt. |
| **Linienstil** | Es werden nur durchgezogene Linien unterstützt. |
| **Drehung** | Es werden Drehungen um 90°, 180° und 270° unterstützt. |

# Intermec IPL

## Textausgabe – Intermec IPL

### Skalierbare Schriftarten

Die folgenden skalierbaren Schriftarten werden unterstützt:

- Century Schoolbook BT
- Dutch 801 Roman BT
- Letter Gothic 12 Pitch BT
- Monospace 821 BT
- OCR-B 10 Pitch BT
- Swiss 721 Bold BT
- Swiss 721 Condensed BT
- Dutch 801 Bold BT
- Futura Light BT
- Monospace 821 Bold BT
- OCR-A BT
- Prestige 12 Pitch Bold BT
- Swiss 721 BT
- Zurich Extra Condensed Bold

Dies sind die einzigen Schriftarten, die dem Benutzer WYSIWYG bieten.

### Adobe-Schriftarten

Alle Adobe-Schriftarten werden den oben genannten Schriftarten zugeordnet (Schriftarten mit ähnlichen Metriken).

### ASCII

Es wird nur die US-amerikanische ASCII-Tabelle unterstützt.

# Barcodeunterstützung – Intermec IPL

Folgende Barcodes werden unterstützt:

- Aztec
- Codabar
- Code 11
- Code 128
- Code 2 of 5 Industrial
- Code 2 of 5 Interleaved
- Code 3 of 9
- Code 49
- Code 93
- DataMatrix
- EAN13
- EAN8
- PDF417
- Planet Code
- QR Code
- RSS14
- RSS14 Expanded
- RSS14 Limited
- RSS14 Stacked
- RSS14 Stacked Omnidirectional
- RSS14 Truncated
- UPC-A
- UPC-E
- UPS Maxicode
- US Postal DPBC
- US Postal Standard
- US Postal Zip-5
-

# RFID-Unterstützung – Intermec IPL

RFID-Barcode wird unterstützt.

| | |
|---|---|
| **Funkprotokoll** | EPC Class 1 Generation 2. |
| **Barcodewert** | Behandelt als HEX96. |

# Unterstützung von Benutzeroberflächenobjekten – Intermec IPL

Unterstützte Benutzeroberflächenobjekte sind nachfolgend aufgelistet.

| Objekt | Kommentar |
|---|---|
| **Vertikale Linien** | Unterstützt. |
| **Horizontale Linien** | Unterstützt. |
| **Diagonale Linien** | Nicht unterstützt. |
| **Rechtecke** | Unterstützt. |
| **Kreise** | Nicht unterstützt. |
| **Bilder** | Unterstützt. |
| **Linienstil** | Es werden nur durchgezogene Linien unterstützt. |
| **Drehung** | Es werden Drehungen um 90°, 180° und 270° unterstützt. |

# Printronix PGL/IGP

## Textausgabe – Printronix PGL/IGP

### Skalierbare Schriftarten

Die folgenden skalierbaren Schriftarten werden unterstützt:

- CG Triumvirate Bold Condensed
- Courier Bold
- Letter Gothic Bold

Dies sind die einzigen Schriftarten, die dem Benutzer WYSIWYG bieten.

### Adobe-Schriftarten

Alle Adobe-Schriftarten werden den oben genannten Schriftarten zugeordnet (Schriftarten mit ähnlichen Metriken).

### ASCII

Es wird nur die US-amerikanische ASCII-Tabelle unterstützt.

## Barcodeunterstützung – Printronix PGL/IGP

Folgende Barcodes werden unterstützt:

- AUSPOST Custom 2
- AUSPOST Custom 3
- AUSPOST Replay Paid
- AUSPOST Standard
- Codabar
- Code 128
- Code 2 of 5 Industrial
- Code 2 of 5 Matrix
- Code 3 of 9
- Code 93
- DataMatrix
- EAN13
- EAN8
- MSI
- PDF417
- Planet Code
- Plessey
- UK/Royal Mail RM4SCCC
- UPC-A
- UPC-E
- UPS Maxicode
- US Postal DPBC
- US Postal Standard
- US Postal Zip-5

# RFID-Unterstützung – Printronix PGL/IGP

RFID-Barcode wird unterstützt.

| | |
|---|---|
| **Funkprotokoll** | EPC Class 1 Generation 2. |
| **Barcodewert** | Behandelt als HEX96. |

# Unterstützung für Benutzeroberflächenobjekte – Printronix PGL/IGP

Unterstützte Benutzeroberflächenobjekte sind nachfolgend aufgelistet.

| **Objekt** | **Kommentar** |
|---|---|
| **Vertikale Linien** | Unterstützt. |
| **Horizontale Linien** | Unterstützt. |
| **Diagonale Linien** | Unterstützt. |
| **Rechtecke** | Unterstützt. |
| **Kreise** | Unterstützt. |
| **Bilder** | Unterstützt. |
| **Linienstil** | Es werden nur durchgezogene Linien unterstützt. |
| **Drehung** | Es werden Drehungen um 90°, 180° und 270° unterstützt. |

# TEC

## Textausgabe – TEC

### Bitmap-Schriftarten

Die folgenden Bitmap-Schriftarten werden unterstützt:

- Courier
- Helvetica
- Letter Gothic
- OCR-A
- OCR-B
- Presentation
- Prestige Elite
- Times New Roman

Dies sind die einzigen Schriftarten, die dem Benutzer WYSIWYG bieten.

### Adobe-Schriftarten

Alle Adobe-Schriftarten werden den oben genannten Schriftarten zugeordnet (Schriftarten mit ähnlichen Metriken).

### ASCII

Es wird nur die US-amerikanische ASCII-Tabelle unterstützt.

## Barcodeunterstützung – TEC

Folgende Barcodes werden unterstützt:

- Code 128
- Code 2 of 5 Industrial
- Code 2 of 5 Interleaved
- Code 2 of 5 Matrix
- Code 3 of 9
- Code 93
- DataMatrix
- EAN13
- EAN8
- MSI
- PDF417
- QR Code
- RSS14
- RSS14 Expanded
- RSS14 Limited
- RSS14 Stacked
- RSS14 Stacked Omnidirectional
- UK/Royal Mail RM4SCCC
- UPC-A
- UPC-E
- UPS Maxicode
- US Postal DPBC
- US Postal Standard
- US Postal Zip-5

## RFID-Unterstützung – TEC

RFID wird nicht implementiert.

## Unterstützung für Benutzeroberflächenobjekte – TEC

Unterstützte Benutzeroberflächenobjekte sind nachfolgend aufgelistet.

| Objekt | Kommentar |
|---|---|
| **Vertikale Linien** | Unterstützt. |
| **Horizontale Linien** | Unterstützt. |
| **Diagonale Linien** | Unterstützt. |
| **Rechtecke** | Unterstützt. |
| **Kreise** | Unterstützt. |
| **Bilder** | Unterstützt. |
| **Linienstil** | Es werden nur durchgezogene Linien unterstützt. |
| **Drehung** | Es werden Drehungen um 90°, 180° und 270° unterstützt. |

# Referenz für die Benutzeroberfläche

**In diesem Kapitel behandelte Themen**


- *LiveCycle Designer ES2 Process-Tool* auf Seite 84
- *Design Center* auf Seite 91

# LiveCycle Designer ES2 Process-Tool

Beim Öffnen eines Adobe LiveCycle Designer ES2-Process wird das Startfenster des Process-Tools angezeigt.

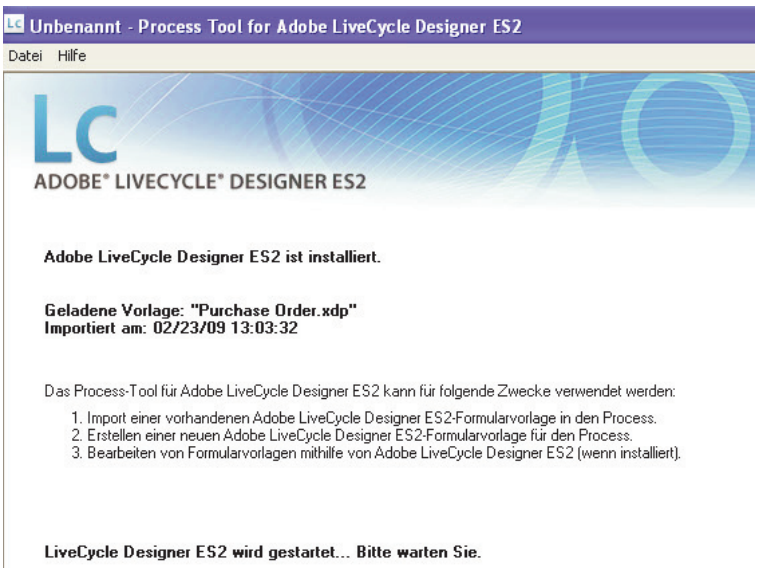

Unbenannt - Process Tool for Adobe LiveCycle Designer ES2
Datei Hilfe
LC
ADOBE® LIVECYCLE® DESIGNER ES2
Adobe LiveCycle Designer ES2 ist installiert.
Geladene Vorlage: "Purchase Order.xdp"
Importiert am: 02/23/09 13:03:32
Das Process-Tool für Adobe LiveCycle Designer ES2 kann für folgende Zwecke verwendet werden:
1. Import einer vorhandenen Adobe LiveCycle Designer ES2-Formularvorlage in den Process.
2. Erstellen einer neuen Adobe LiveCycle Designer ES2-Formularvorlage für den Process.
3. Bearbeiten von Formularvorlagen mithilfe von Adobe LiveCycle Designer ES2 (wenn installiert).
LiveCycle Designer ES2 wird gestartet... Bitte warten Sie.

*Abbildung 5 Startfenster des Process-Tools für Adobe LiveCycle Designer ES2*

Wenn eine Vorlage geladen wird, werden der Name der Vorlage und das Datum des Vorlagenimports angezeigt.

Wenn LiveCycle Designer ES2:

- installiert ist und Sie festgelegt haben, dass die LiveCycle Designer-Benutzeroberfläche beim Start geladen werden soll, wird die LiveCycle Designer-Benutzeroberfläche gestartet.
- nicht installiert ist, können Sie Ihre Einstellungen über die Befehle im Menü **Datei** in diesem Fenster vornehmen und die gewünschte Vorlage auswählen.

### In diesem Kapitel behandelte Themen

- *Befehle im Menü „Datei“* auf Seite 85
- *Dialogfeld „Einstellungen“* auf Seite 86
- *Dialogfeld „Vorlage auswählen“* auf Seite 89

# Befehle im Menü „Datei“

| | |
|---|---|
| **Vorlage öffnen** | Öffnet das *Dialogfeld „Vorlage auswählen“*, in dem Sie eine Vorlage auswählen können. |
| **Speichern** | Speichert die vorgenommenen Änderungen. |
| **LiveCycle-Archiv importieren** | Öffnet das Dialogfeld **Zu importierendes LCA-Archiv auswählen**, in dem Sie die zu importierende LiveCycle Archive(LCA)-Datei auswählen können. Die LCA-Datei enthält ein vollständiges Paket aus XDP-Hauptvorlage und den zugehörigen Abhängigkeiten.<br>Um das Paket in das Process-Tool importieren zu können, wurde die LCA-Datei zuvor als Ressourcentyp **Muster** in Design Center importiert. |
| **Message-Schema exportieren** | Erstellt eine XML-Schemadatei, die die StreamServe-Message enthält. |
| **XML-Vorschau exportieren** | Erstellt eine XML-Datei, die die Struktur der StreamServe-Message enthält, mit Musterdaten für jedes Feld. |
| **Ressourcen in Datei extrahieren** | Öffnet das Dialogfeld **Zu extrahierende Ressourcen auswählen**, in dem Sie das Ressourcenset und dann die in die Datei zu extrahierenden Ressourcen auswählen können. Die extrahierten Ressourcen werden im aktuellen Arbeitsverzeichnis des Process-Tools, mit der gleichen Struktur wie im Ressourcenset, gespeichert und stehen dann in LiveCycle Designer zur Verfügung. |
| **Einstellungen** | Öffnet das *Dialogfeld „Einstellungen“*. |
| **Beenden** | Beendet LiveCycle Designer. Wenn geänderte oder noch nicht gespeicherte Projektkomponenten vorhanden sind, wird ein Dialogfeld geöffnet, in dem Sie die zu speichernden Elemente angeben können. |

# Dialogfeld „Einstellungen“

Im Dialogfeld **Einstellungen** können Sie Standardeinstellungen definieren und StreamServe-Variablen SOM-Ausdrücke zuordnen.

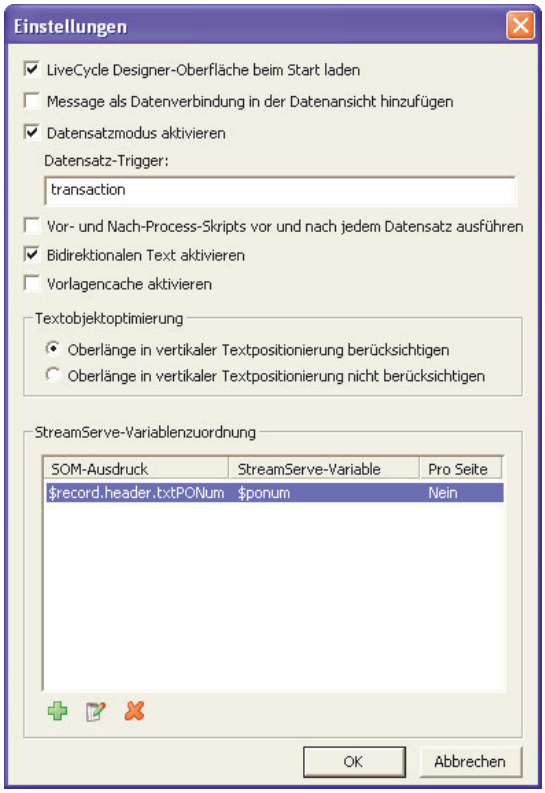



*Abbildung 6 Dialogfeld „Einstellungen“*

### LiveCycle Designer-Benutzeroberfläche beim Start laden

Aktivieren Sie diese Option, wenn Sie die Benutzeroberfläche von LiveCycle Designer ausführen möchten. Die Benutzeroberfläche wird automatisch beim Starten des Process-Tools geöffnet.

**Hinweis:** Wenn Sie LiveCycle Designer ES2 nicht installiert haben, steht diese Option nicht zur Verfügung.

### Message als Datenverbindung in der Datenansicht hinzufügen

Aktivieren Sie diese Option, wenn Sie eine Vorlage von Grund auf neu erstellen möchten. Die StreamServe-Message wird LiveCycle Designer als Datenverbindung hinzugefügt.

### Datensatzmodus aktivieren

Der Process kann in zwei Modi ausgeführt werden, im Datensatzmodus und im Nicht-Datensatzmodus. Im Datensatzmodus wird das Datendokument als Sequenz von Datensätzen behandelt. Im einfachsten Fall wird jeweils ein Datensatz geladen, verarbeitet und dann wieder entladen, bevor der nächste Datensatz geladen wird. Hauptzweck des Datensatzmodus ist es, den Ressourcenverbrauch (Arbeitsspeicher und CPU-Zyklen) bei der Verarbeitung großer Datendokumente zu verringern.

Ausreichende Ressourcen vorausgesetzt, können alle im Datensatzmodus vorgenommenen Schritte auch im Nicht-Datensatzmodus durchgeführt werden.

### Datensatz-Trigger

Wenn die Option **Datensatzmodus aktivieren** ausgewählt ist, ist dieses Feld aktiviert. Das in diesem Feld angegebene XML-Tag oder die XML-Ebene (eine positive Zahl) legt fest, auf welche Weise die Eingabedaten in mehrere Datensätze aufgeteilt werden sollen.

*Beispiel 5* *Datensatz-Trigger mit folgender Eingabedatenstruktur*

```
<A>
 <B>
   <Data1>1</Data1>
   <Data2>2</Data2>
 </B>
 <C>
   <Data1>3</Data1>
   <Data2>4</Data2>
 </C>
 <B>
   <Data1>5</Data1>
   <Data2>6</Data2>
 </B>
</A>
```

Wenn Sie **Datensatz-Trigger** auf

- `1` setzen, gibt es drei Datensätze.
- XML-Tag `B` setzen, gibt es zwei Datensätze (die vom `B`-Tag eingeschlossenen). Der vom `C`-Tag eingeschlossene Datensatz wird ignoriert.

  Wenn Sie ein XML-Tag als Datensatz-Trigger verwenden, wird die Tiefensuche ausgeführt. Das bedeutet, dass das zuerst gefundene XML-Tag zusammen mit der zugehörigen Ebene für die weitere Suche verwendet wird. Es werden nur Datensätze berücksichtigt, die von dem angegebenen XML-Tag umschlossen sind und sich auf derselben Ebene befinden wie der zuerst gefundene Datensatz.

### Vor- und Nach-Process-Skripts vor und nach jedem Datensatz ausführen

Wenn die Option **Datensatzmodus aktivieren** ausgewählt ist, ist dieses Feld aktiviert.

In diesem Fall werden Vor- und Nach-Process-Skripts vor und nach jedem Datensatz ausgeführt. Andernfalls erfolgt die Ausführung vor und nach jedem Process.

Bei Projekten, die in LiveCycle Production Print ES2 erstellt werden, ist diese Funktionalität standardmäßig aktiviert.

In vorhandenen Projekten aus Versionen vor LiveCycle Production Print ES2 werden die Skripts dagegen standardmäßig Process-bezogen ausgeführt.

Informationen hierzu finden Sie unter *Verwenden von Vor- und Nach-Process-Skripts* auf Seite 106.

### Bidirektionalen Text aktivieren

Mit dieser Option aktivieren Sie bidirektionalen Text (BiDi). Bidirektionaler Text kann für die arabische und die hebräische Notation verwendet werden. Die Option **Bidirektionalen Text aktivieren** ist standardmäßig deaktiviert.

### Vorlagencache aktivieren

Wählen Sie diese Option, um das Zwischenspeichern von Vorlagen zu aktivieren. Bei der Verwendung dynamischer Vorlagen können Sie so die Leistung signifikant steigern.

Informationen hierzu finden Sie unter *Vorlagenzwischenspeicherung* auf Seite 12.

### Textobjektoptimierung

Der Ascent-Wert einer Schriftart gibt die Höhe eines Zeichens oberhalb der Grundlinie an. Dieser Wert ist in der Regel für alle Zeichen einer Schriftart identisch. Wenn Sie jedoch Sonderzeichen (wie z. B. die schwedischen Zeichen Ä und Å) verwenden, überschreiten diese möglicherweise den Ascent-Wert für die jeweilige Schriftart, d. h. die Zeichenhöhe. Um sich überlappende Textzeilen zu vermeiden, können Sie der Textzeilenhöhe den Oberlängenwert hinzufügen.

- **Oberlänge in vertikaler Textpositionierung berücksichtigen**
  Die Oberlänge wird berechnet, und der entsprechende Wert wird der Textzeilenhöhe für die Schriftart hinzugefügt. Da hierbei jedes einzelne Zeichen der Schriftart berücksichtigt wird, hat dies Auswirkungen auf die Leistung. Wählen Sie diese Option nicht, wenn Sie keine Sonderzeichen verwenden. Die Option **Oberlänge in vertikaler Textpositionierung berücksichtigen** ist standardmäßig aktiviert.
- **Oberlänge in vertikaler Textpositionierung nicht berücksichtigen**
  Die Oberlänge wird nicht berücksichtigt. Wenn Sie Sonderzeichen verwenden, kann es zur Überlappung der Textzeilen kommen. Aktivieren Sie diese Option, wenn für Sie die Leistung Priorität hat.

### StreamServe-Variablenzuordnung

Dieser Liste können Sie die Zuordnung zwischen dem **SOM-Ausdruck** und der **StreamServe-Variablen** entnehmen.

Durch die Option **Pro Seite** wird angegeben, ob die Zuordnung dokument- oder seitenbezogen erfolgt:

| | |
|---|---|
| `Ja` | Die Zuordnung erfolgt einmal vor Ausgabe jeder Seite. |
| `Nein` | Die Zuordnung erfolgt einmal vor Ausgabe jedes Dokuments. |

Die seitenbezogene Zuordnung wird im Dialogfeld **Variablen zuordnen** durch die Option **Variable für jede Seite auswerten** angegeben.

Informationen hierzu finden Sie unter *Übernehmen eines SOM-Ausdruckswerts in eine StreamServe-Variable* auf Seite 102.

## Dialogfeld „Vorlage auswählen“

Wählen Sie im Dialogfeld **Vorlage auswählen** eine Vorlage aus, und geben Sie die entsprechenden Verbindungsdetails an.

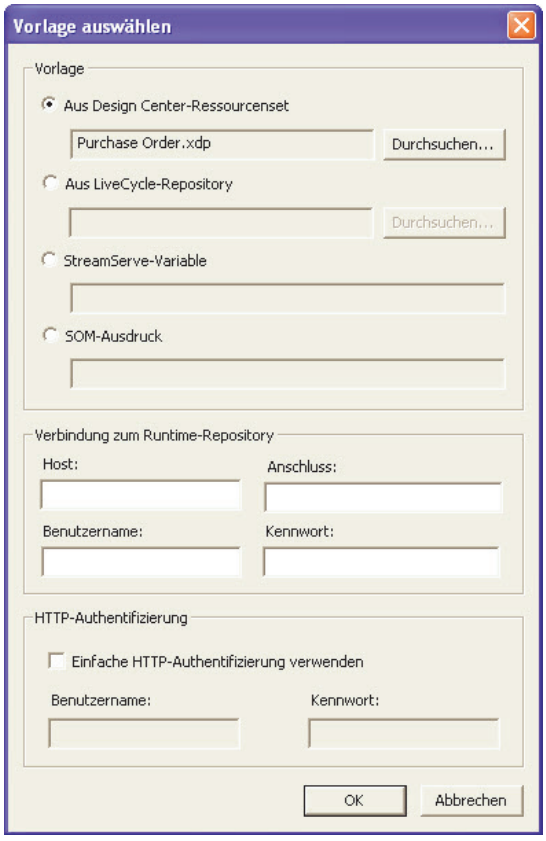



*Abbildung 7 Dialogfeld „Vorlage auswählen“*

Die Einstellungen werden in der folgenden Tabelle beschrieben:

| Einstellungen | Beschreibung |
|---|---|
| **Aus Design Center-Ressourcenset** | Wählen Sie eine in einem Ressourcenset enthaltene Vorlage aus. Die ausgewählte Vorlage wird statisch mit dem Process verknüpft. |
| **Aus LiveCycle-Repository** | Navigieren Sie im Repository zu der gewünschten Vorlage, und wählen Sie sie aus. Es werden die Verbindungsdaten und Anmeldeinformationen für das Repository gemäß der Angabe unter **Verbindung zum Runtime-Repository** verwendet.<br>Die ausgewählte Vorlage wird zur Laufzeit beim Starten des Process aus dem Repository geladen. |
| **StreamServe-Variable** | Geben Sie eine StreamServe-Variable an, die auf eine Vorlage verweist.<br>Informationen hierzu finden Sie unter *Verwenden einer StreamServe-Variablen zum Laden der Vorlage* auf Seite 24. |
| **SOM-Ausdruck** | Geben Sie einen SOM-Ausdruck im Daten-DOM an, der auf eine Vorlage verweist. Beispiel: `$record.templateLocation`<br>Informationen hierzu finden Sie unter *Verwenden eines SOM-Ausdrucks zum Laden einer Vorlage* auf Seite 26. |
| **Verbindung zum Runtime-Repository** | Geben Sie das Verbindungsprofil für das LiveCycle-Repository an. |
| **HTTP-Authentifizierung** | Wenn Sie mit einer HTTP-URI arbeiten, können Sie die Option **Einfache HTTP-Authentifizierung verwenden** auswählen. In diesem Fall erfolgt die Anmeldung über den angegebenen **Benutzernamen** mit dem dazugehörigen **Kennwort**. |

# Design Center

In diesem Kapitel werden die speziellen Design Center-Befehle für LiveCycle Production Print ES2 beschrieben.

### In diesem Kapitel behandelte Themen


- *Befehle im Menü „Extras“* auf Seite 91
- *Befehle im Menü „Ressourcen“* auf Seite 91
- *Dialogfelder* auf Seite 92


## Befehle im Menü „Extras“

Im Menü **Extras** sind die in der folgenden Tabelle beschriebenen Befehle enthalten.

| | |
|---|---|
| **Adobe LiveCycle-Repositoryverbindung auswählen** | Öffnet das *Dialogfeld „Aktive Adobe LiveCycle-Repositoryverbindung auswählen“*, in dem Sie die Verbindungen zum LiveCycle ES2-Repository verwalten können. |

## Befehle im Menü „Ressourcen“

Bei aktiver Ressourcenset-Ansicht ist das Menü **Ressourcen** verfügbar. Darin sind die in der folgenden Tabelle beschriebenen Befehle enthalten.

| | |
|---|---|
| **Aus Adobe LiveCycle Repository importieren** | Öffnet das Dialogfeld **Ressource auswählen**. In diesem Dialogfeld können Sie eine Ressource im LiveCycle ES2-Repository auswählen. Nach Auswahl einer Ressource wird das *Dialogfeld „Adobe LiveCycle-Repositoryressource importieren“* geöffnet. |
| **Alle Adobe LiveCycle-Ressourcen aktualisieren** | Öffnet das *Dialogfeld „Alle Adobe LiveCycle-Repositoryressourcen aktualisieren“*. |

# Dialogfelder

## Dialogfeld „Aktive Adobe LiveCycle-Repositoryverbindung auswählen“

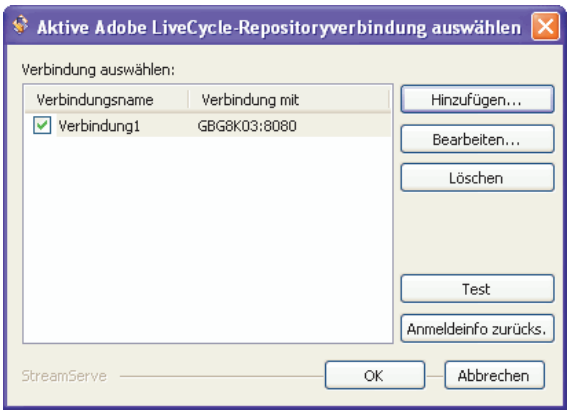



*Abbildung 8 Dialogfeld „Aktive Adobe LiveCycle-Repositoryverbindung auswählen“*

Dieses Dialogfeld enthält eine Liste der verfügbaren Verbindungen zu LiveCycle ES2-Repositorys.

**Hinweis:** Sie können Verbindungen zu verschiedenen Repositorys definieren, jedoch jeweils nur Ressourcen für ein Repository erstellen und aktualisieren.

Die ausgewählte Verbindung ist die aktive Verbindung.

| Option | Beschreibung |
|---|---|
| **Verbindung auswählen:** | Alle definierten Verbindungen zu LiveCycle ES2-Repositorys.<br>• **Verbindungsname**: Der Name der Verbindung.<br>• **Verbindung mit**: Der Name oder die IP-Adresse des verbundenen Hosts und die Anschlussnummer. |
| **Hinzufügen** | Fügt eine neue Verbindung hinzu. |
| **Bearbeiten** | Ermöglicht das Bearbeiten einer vorhandenen Verbindung. |
| **Löschen** | Löscht die ausgewählte (markierte) Verbindung. |
| **Test** | Überprüft, ob die ausgewählte Verbindung erfolgreich hergestellt werden kann. |

| Option | Beschreibung |
|---|---|
| **Anmeldeinformationen zurücksetzen** | Löscht die Anmeldeinformationen für ein Adobe LiveCycle-Repository. Verwenden Sie diese Einstellung, wenn sich der Benutzername oder das Kennwort auf dem Server geändert hat oder wenn Sie das Kennwort falsch eingegeben haben. |

## Dialogfeld „Adobe LiveCycle-Repositoryressource importieren“

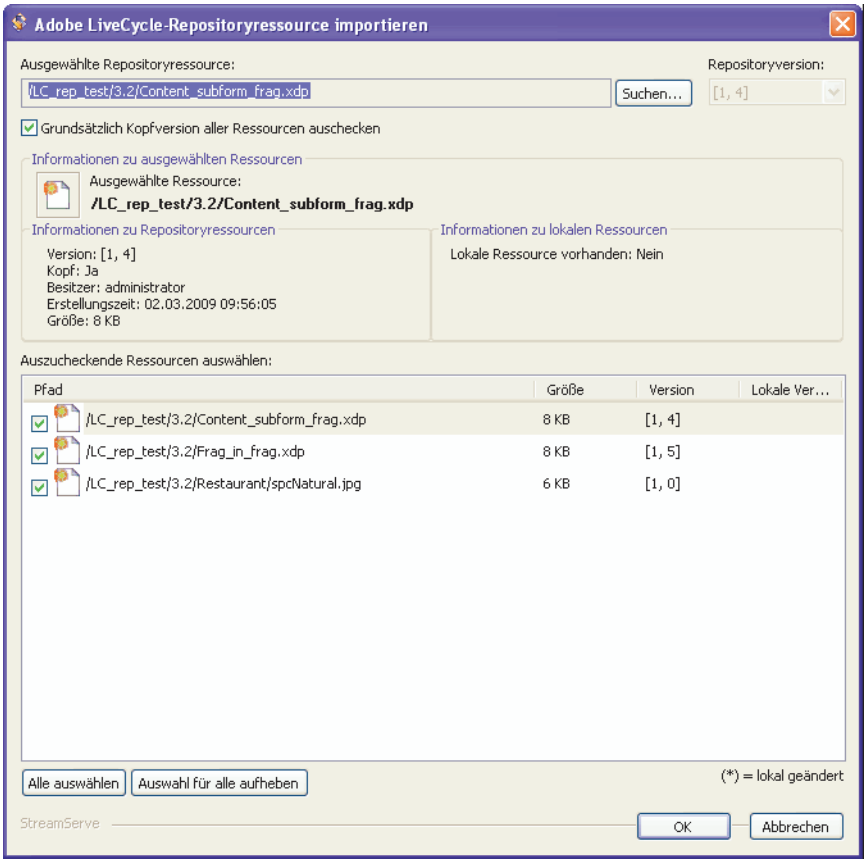



*Abbildung 9 Dialogfeld „Adobe LiveCycle-Repositoryressource importieren“*

Informationen zu den in diesem Dialogfeld verwendeten Dokumentsymbolen finden Sie unter *Beim Zugriff auf das LiveCycle ES2-Repository verwendete Symbole* auf Seite 99.

Die Einstellungen werden in der folgenden Tabelle beschrieben.

| Option | Beschreibung |
|---|---|
| **Ausgewählte Repository-ressource** | Zeigt den Pfad und Dateinamen der ausgewählten Ressource an. |
| **Repositoryversion** | Ressourcenversion im Repository. Die verfügbaren Ressourcen können über die Dropdownliste ausgewählt werden. |
| **Grundsätzlich Kopfversion aller Ressourcen auschecken** | Importiert die aktuelle Version der Ressource und die zugehörigen Abhängigkeiten. Diese Option ist standardmäßig aktiviert. |
| **Informationen zu Repositoryressourcen** | Zeigt Informationen zur ausgewählten (in der Liste markierten) Ressource im Repository an:<br>**Version**: Versionsnummer der ausgewählten Ressource.<br>**Kopf**: Ja/Nein<br>**Ja**, wenn die aktuelle Version die neueste ist. Andernfalls **Nein**.<br>**Besitzer**: Benutzername des Erstellers.<br>**Erstellungszeit**: Datum und Uhrzeit, zu der die ausgewählte Ressourcenversion erstellt wurde.<br>**Größe**: Größe der Ressourcendatei (KB).<br>Wenn die Ressource aus dem Repository entfernt wurde, werden entsprechende Informationen angezeigt. |
| **Informationen zu lokalen Ressourcen** | Zeigt Informationen über die ausgewählte (markierte) Ressource im lokalen Ressourcenset an:<br>**Lokale Ressource vorhanden**: Ja/Nein.<br>**Ja**, wenn die Ressource bereits lokal im Design Center-Ressourcenset vorhanden ist.<br>**Lokale Version**: Die Versionsnummer der Ressource, die lokal im Design Center-Ressourcenset gespeichert ist.<br>**Importiert**: Datum und Uhrzeit, zu der die Ressource im Design Center-Ressourcenset importiert und gespeichert wurde.<br>**Lokal geändert**: Ja/Nein.<br>**Ja**, wenn die Ressource lokal geändert wurde. |

| Option | Beschreibung |
|---|---|
| **Auszucheckende Ressourcen auswählen** | Eine Liste ausgewählter Repositoryressourcen und der zugehörigen Abhängigkeiten.<br>**Pfad**: Pfad und Dateiname der Ressource.<br>**Größe**: Größe der Ressourcendatei (KB).<br>**Version**: Versionsnummer der Ressource im Repository.<br>**Lokale Version**: Versionsnummer der im Design Center-Ressourcenset gespeicherten Ressource. |
| **Alle auswählen** | Wählt alle Objekte in der Liste aus. |
| **Auswahl aufheben** | Hebt die Auswahl für alle Kontrollkästchen auf (d. h., keines der Objekte in der Liste ist ausgewählt). |

## Dialogfeld „Alle Adobe LiveCycle-Repositoryressourcen aktualisieren“

In diesem Dialogfeld wird angezeigt, ob sich eine Ressource geändert hat und aktualisiert werden muss.

Beim Öffnen des Dialogfelds sind die Ressourcen, deren Version im Design Center-Ressourcenset älter ist als im Repository, bereits ausgewählt.

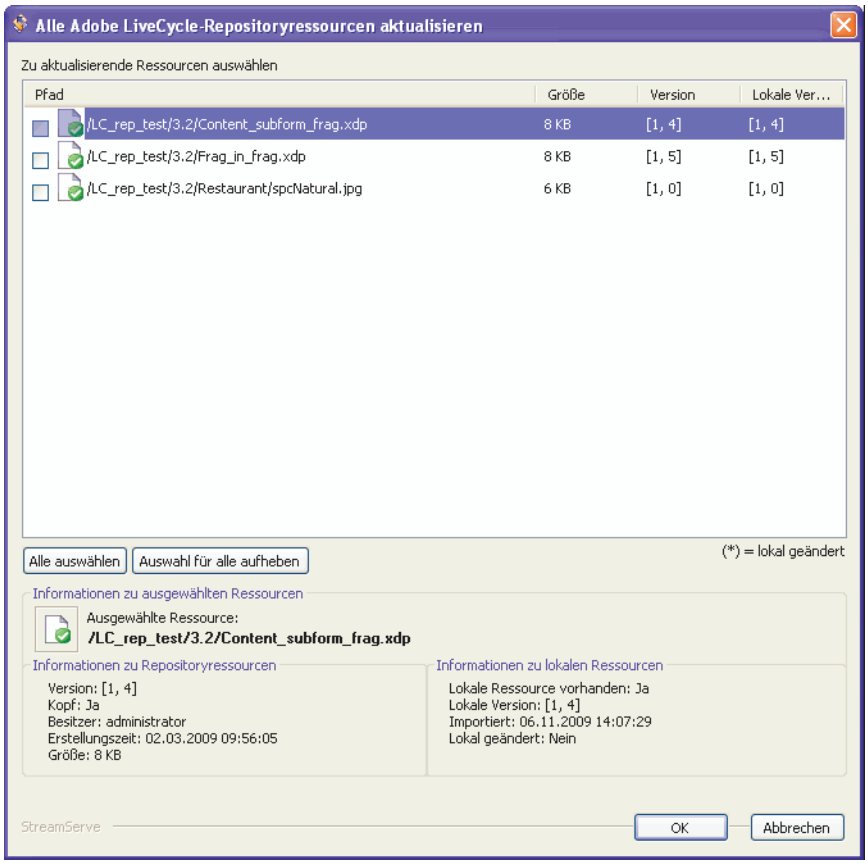



*Abbildung 10 Dialogfeld „Alle Adobe LiveCycle-Repositoryressourcen aktualisieren“*

Informationen zu den in diesem Dialogfeld verwendeten Dokumentsymbolen finden Sie unter *Beim Zugriff auf das LiveCycle ES2-Repository verwendete Symbole* auf Seite 99.

| Option | Beschreibung |
|---|---|
| **Zu aktualisierende Ressourcen auswählen** | **Pfad**: Pfad und Dateiname der Ressource.<br>**Größe**: Größe der Ressourcendatei.<br>**Version**: Version der Ressource im Repository.<br>**Lokale Version**: Version der im Design Center-Ressourcenset gespeicherten Ressource. |
| **Alle auswählen** | Wählt alle Objekte in der Liste aus. |
| **Auswahl aufheben** | Hebt die Auswahl für alle Kontrollkästchen auf (d. h., keines der Objekte in der Liste ist ausgewählt). |

| Option | Beschreibung |
|---|---|
| **Informationen zu ausgewählten Ressourcen** | Informationen zur ausgewählten (markierten) Ressource. |
| **Informationen zu Repository-ressourcen** | Zeigt Informationen zur Ressource im Repository an:<br>**Version**: Version der ausgewählten Ressource.<br>**Kopf**: **Ja**/**Nein**<br>**Ja**, wenn die aktuelle Version die neueste ist.<br>**Besitzer**: Benutzername des Erstellers.<br>**Erstellungszeit**: Datum und Uhrzeit, zu der die ausgewählte Ressourcenversion erstellt wurde.<br>**Größe**: Größe der Ressourcendatei.<br>Wenn die Ressource aus dem Repository entfernt wurde, werden entsprechende Informationen angezeigt. |
| **Informationen zu lokalen Ressourcen** | **Lokale Ressource vorhanden**: **Ja**/**Nein**. **Ja**, wenn die Ressource in das Design Center-Ressourcenset importiert wurde.<br>**Lokale Version**: Version der im Design Center-Ressourcenset gespeicherten Ressource.<br>**Importiert**: Datum und Uhrzeit, zu der die Ressource im Design Center-Ressourcenset importiert und gespeichert wurde.<br>**Lokal geändert**: **Ja**/**Nein**.<br>**Ja**, wenn die Ressource lokal geändert wurde. |

# Dialogfeld „Runtime-Process-Einstellungen“ – Registerkarte „Allgemein“

Im Dialogfeld **Runtime-Process-Einstellungen** können Sie Runtime-spezifische Einstellungen für den ausgewählten Process konfigurieren.

Auf der Registerkarte **Allgemein** legen Sie die allgemeinen Einstellungen für den Runtime-Process fest.

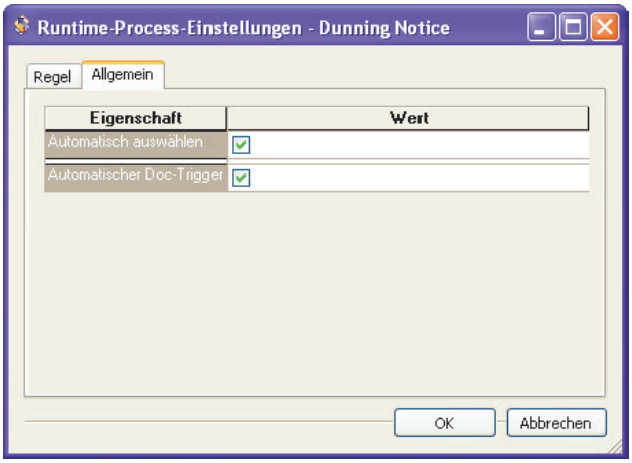



*Abbildung 11 Runtime-Process-Einstellungen*

| Option | Beschreibung |
|---|---|
| **Automatisch auswählen** | Weitere Informationen finden Sie in der Standard-dokumentation zu *StreamServe Design Center*. |
| **Automatischer Doc-Trigger** | Wenn Sie diese Option auswählen, ist die **Document-Triggervariable** (angegeben auf der Registerkarte **Document-Trigger** des Dialogfelds **Runtime-Connector-Einstellungen**) deaktiviert. Damit wird jeder Datensatz automatisch einem Dokument zugeordnet. Eine Gruppierung der Ausgabe in logische Dokumente erfolgt in diesem Fall nicht.<br><br>Bei aktualisierten Projekten aus Versionen vor LiveCycle Designer ES2 ist diese Einstellung standardmäßig ausgewählt.<br><br>Bei neuen LiveCycle Designer ES2-Processes ist sie dagegen nicht ausgewählt. |

Die Einstellung **Automatischer Doc-Trigger** ist auch für StoryTeller-Processes verfügbar. In diesem Fall wird jeder Process automatisch einem Dokument zugeordnet.

# Beim Zugriff auf das LiveCycle ES2-Repository verwendete Symbole

In den Dialogfeldern werden verschiedene Dokumentsymbole verwendet:

Die im LiveCycle ES2-Repository gespeicherte Ressource hat eine aktuellere Version als die lokale Ressource (im Design Center-Ressourcenset). Beim Importieren wird die lokale Ressource mit der aktuelleren Version überschrieben.

Die Ressource ist im Design Center-Ressourcenset nicht vorhanden. Sie wird beim Importieren erstellt.

Die Ressource im lokalen Design Center-Ressourcenset hat dieselbe Version wie die Ressource im LiveCycle ES2-Repository. Es ist kein Importvorgang erforderlich.

Die lokale Ressource (im Design Center-Ressourcenset) hat eine aktuellere Version als die im LiveCycle ES2-Repository gespeicherte Ressource. Beim Importieren wird die lokale Ressource mit einer älteren Version überschrieben.

Die Ressource wurde aus dem LiveCycle ES2-Repository entfernt, ist aber lokal noch vorhanden.

# Skripts

### In diesem Kapitel behandelte Themen


- *Übernehmen eines SOM-Ausdruckswerts in eine StreamServe-Variable* auf Seite 102
- *Zugriff auf StreamServe-Variablen im XFA-Prozessor* auf Seite 105
- *Verwenden von Vor- und Nach-Process-Skripts* auf Seite 106
- *Schreiben in die StreamServe-Protokolldatei* auf Seite 108

# Übernehmen eines SOM-Ausdruckswerts in eine StreamServe-Variable

Sie können die Werte aus einem SOM-Ausdruck einer StreamServe-Variable zuordnen.

Auf diese Weise ist es möglich, Daten aus LiveCycle Designer zu verwenden, um beispielsweise die Nachbearbeitung zu steuern oder Treiberoptionen entsprechend der Eingabe, der Vorlage, dem Formular, dem Layout oder den Skriptdaten festzulegen.

Beachten Sie, dass diese Zuordnung nur in eine Richtung funktioniert. Während also Werte aus einem SOM-Ausdruck in eine StreamServe-Variable übernommen werden können, ist dies umgekehrt nicht möglich.

Die Zuordnung kann einmal pro Dokument oder einmal pro Seite erfolgen:

- Pro Dokument: Die Werte werden genau einmal für jedes Dokument zugeordnet. Die Zuordnung erfolgt vor Ausgabe des Dokuments, unmittelbar nachdem alle Seiten produziert wurden.
- Pro Seite: Die Zuordnung vor Ausgabe jeder Seite. Das ermöglicht die Nutzung erweiterter Blattlayout- und Document Broker-Funktionen. So können Sie beispielsweise mithilfe eines aus einem SOM-Ausdruck extrahierten seitenbezogenen Werts echte OMR-Codes für die AFP-Ausgabe erstellen.

  Die Zuordnung erfolgt zunächst einmal für das gesamte Dokument und anschließend nacheinander für jede Seite, bevor diese ausgegeben wird.

  **Hinweis:** Für die seitenbezogene Zuordnung werden relative SOM-Ausdrücke verwendet, um auf Layoutobjekte Bezug zu nehmen. Da relative SOM-Ausdrücke im Seitenkontext ausgewertet werden, sind unterschiedliche Wert für verschiedene Seiten möglich. Objekte ohne Seitenkontext werden im Dokumentkontext ausgewertet und erhalten identische Werte für alle Seiten.

*Beispiel 6* *SOM-Ausdruck und StreamServe-Variable*

SOM-Ausdruck: `$record.header.txtPONum`

StreamServe-Variable: `$ponum`

*Beispiel 7* *Zuordnen eines Werts zu einer StreamServe-Variablen*

Eingabedaten für den XFA-Prozessor:

```
<?xml version="1.0" encoding="UTF-8"?>
<batch>
<transaction>
 <header>
  <txtPONum>1234567890</txtPONum>
  <dtmDate>2004-02-08</dtmDate>
  <txtOrderedByCompanyName>Another Company</txtOrderedByCompany-
Name>
  <txtOrderedByAddress>123, Any St.</txtOrderedByAddress>
  …
```

Die Variablenzuordnung weist den Wert des Elements `txtPONum` der StreamServe-Variablen `$ponum` zu:

SOM-Ausdruck: `$record.header.txtPONum`

StreamServe-Variable: `$ponum`

*Beispiel 8* *Seitenbezogene Zuordnung – Relativer SOM-Ausdruck für untergeordnete Objekte der Masterseite*

Relative SOM-Ausdrücke für untergeordnete Objekte der Masterseite müssen bei der ersten Ebene beginnen.

Eine Masterseite (`Page1`) enthält folgende Elemente:

- Textfeld `PageNumber`
- Unterformular `CustomerInfo` mit den Feldern `CustomerName` und `CustomerID`

Der relative SOM-Ausdruck für das Feld `PageNumber` ist "`PageNumber`".

Der relative SOM-Ausdruck für `CustomerName` ist "`CustomerInfo.CustomerName`".

*Beispiel 9* *Seitenbezogene Zuordnung – Relative SOM-Ausdrücke für untergeordnete Objekte des Haupt-Unterformulars*

Relative SOM-Ausdrücke für untergeordnete Objekte des Haupt-Unterformulars müssen bei der ersten Ebene (Seitenebene) beginnen.

Der Verweis auf ein Haupt-Unterformular mit der Bezeichnung `Data` kann über den absoluten SOM-Ausdruck "`$form.Data`" erfolgen.

Das Unterformular `Data` enthält das untergeordnete Formular `Part1`, in dem sich wiederum das Feld `Header1` befindet.

Der relative SOM-Ausdruck für `Header1` lautet demnach "`Part1.Header1`".

Informationen zur Syntax des SOM-Ausdrucks finden Sie in der LiveCycle Designer-Hilfe: *Scripting > Scripting Using LiveCycle Designer ES2 > Referencing Objects in Calculations and Scripts*.

## Informationen zu SOM-Ausdrücken

Das XFA Scripting Object Model (SOM) ist ein Modell zum Verweisen auf Werte, Eigenschaften und Methoden in einem bestimmten Document Object Model (DOM). Ein DOM strukturiert Objekte und Eigenschaften als Hierarchiestruktur. XFA SOM-Ausdrücke ermöglichen durch eine einfache Objektverweissyntax einen problemlosen Zugriff auf diese Objekte und Eigenschaften.

Detaillierte Informationen zur SOM-Spezifikation finden Sie unter XML Forms Architecture (XFA) Specification.

Den SOM-Ausdruck für ein bestimmtes XFA-Objekt können Sie mithilfe der Skripteigenschaft `somExpression` ermitteln. Auf diese Weise kann der vollständige Ausdruck für das Objekt extrahiert werden.

*Beispiel 10* *ECMAscript-Beispiel*

```
this.rawValue = this.somExpression;
```

Wenn Sie dieses Skript dem Objekt zuordnen, für das Sie den SOM-Ausdruck abrufen möchten, wird der dem Wert dieses Objekts zugeordnete SOM-Ausdruck zurückgegeben, und die Zeichenfolge wird mit dem Dokument ausgegeben.

Weitere Informationen zu SOM-Ausdrücken finden Sie in der Dokumentation zu *Adobe LiveCycle Designer ES2*.

# Zugriff auf StreamServe-Variablen im XFA-Prozessor

StreamServe-Variablen können direkt aus einem Skript im XFA-Prozessor gelesen werden. Dies funktioniert, wenn eine StreamServe-Message als Datenverbindung für eine Vorlage verwendet wird (d. h., die Einstellung **Message als Datenverbindung in der Datenansicht hinzufügen** ist aktiviert).

Alle StreamServe-Variablen, die vor dem Ausführen des Process erstellt wurden, können mithilfe der folgenden Syntax über die Datenverbindung gelesen werden:

```
$record.variables.variableid;
```

*Beispiel 11* *StreamServe-Skript vor Process*

```
$myvar = "myvalue";
//Weist "myvalue" der StreamServe-Variable "myvar" zu
```

*Beispiel 12* *Ecma-Skript in Formular auf Feld in Vorlage*

```
this.rawValue = $record.variables.myvar;
//Weist den Wert der StreamServe-Variable "myvar" dem Rohwert des
aktuellen Objekts zu.
```

**Hinweis:** Von SAP- (beispielsweise RDI-Header-Variablen) und Lawson-Agenten erstellte schreibgeschützte Variablen können auf diese Weise nicht gelesen werden. Solche Variablenwerte können erst dann im XFA-Prozessor gelesen werden, wenn ihnen Skriptvariablen (Lesen/Schreiben) zugeordnet wurden.
Ob diese Methode verwendet werden kann, richtet sich nach der Skriptzuordnung und dem Ausführungszeitpunkt. Das Objekt `pageSet` und die abhängigen Elemente können ausschließlich Skriptvariablen im `layout:ready`-Event und im `prePRint`-Event problemlos zugeordnet werden.

# Verwenden von Vor- und Nach-Process-Skripts

Sie können Vor- und Nach-Process-Skripts vor und nach jedem Datensatz ausführen.

Durch das Ausführen von Skripts vor und/oder nach einzelnen Datensätzen wird ein Kontext für aus SOM-Ausdrücken extrahierte Variablenwerte bereitgestellt. Informationen hierzu finden Sie unter *StreamServe-Variablenzuordnung* auf Seite 89.

Die Zuordnung von SOM-Ausdrücken zu StreamServe-Variablen erfolgt datensatzbezogen oder, sofern die Option **Variable für jede Seite auswerten** aktiviert ist, sogar für jede Seite innerhalb eines Datensatzes. Wenn diese Einstellung:

- aktiviert ist, wird das Vor-/Nach-Process-Skript für jeden Datensatz mit diesen Werten aufgerufen,
- deaktiviert ist, werden nur die aus dem letzten Datensatz extrahierten Werte an das Nach-Process-Skript übergeben.

**Hinweis:** Das erste Vor-Process-Skript und das letzte Nach-Process-Skript werden in der Vor-Process-Phase (vor und nach dem eigentlichen Process) sowie in der Process-Phase selbst ausgeführt. Alle anderen Vor- und Nach-Process-Skripts werden dagegen ausschließlich in der Process-Phase ausgeführt.

## Funktionsweise – Reihenfolge der Ausführung von Skripts

In der nachstehenden Abbildung wird dargestellt, in welcher Reihenfolge die Vor- und Nach-Process-Skripts bei aktivierter Option **Variable für jede Seite auswerten** ausgeführt werden.

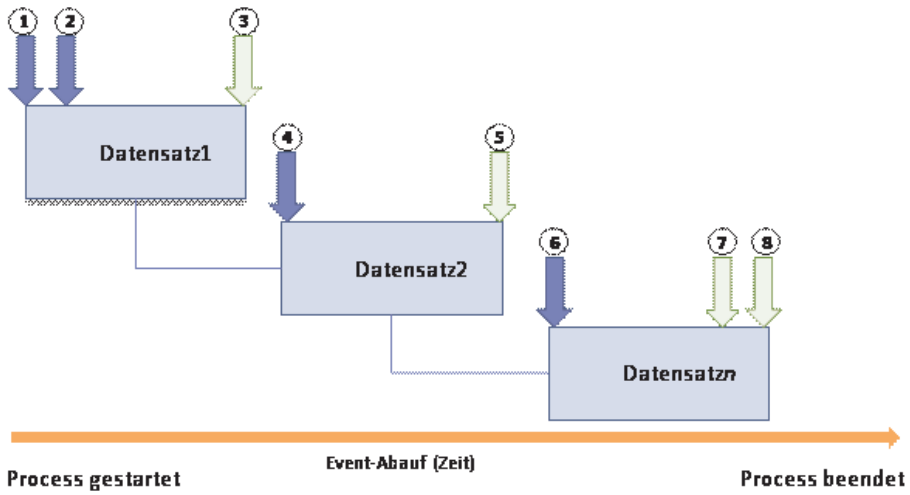



*Abbildung 12 Reihenfolge der Ausführung von Skripts*

1. Das Vor-Process-Skript wird vor Process-Beginn ausgeführt.
2. Das Vor-Process-Skript wird vor dem ersten Datensatz ausgeführt.
3. Das Nach-Process-Skript wird nach dem ersten Datensatz ausgeführt.
4. Das Vor-Process-Skript wird vor allen nachfolgenden Datensätzen ausgeführt.
5. Das Nach-Process-Skript wird nach allen nachfolgenden Datensätzen ausgeführt.
6. Das Vor-Process-Skript wird vor dem letzten Datensatz ausgeführt.
7. Das Nach-Process-Skript wird nach dem letzten Datensatz ausgeführt.
8. Das Nach-Process-Skript wird nach dem Process-Ende ausgeführt.

### So aktivieren Sie Vor- und Nach-Process-Skripts

Mit der Option **Vor- und Nach-Process-Skripts vor und nach jedem Datensatz ausführen** im Dialogfeld **Einstellungen** können Sie diese Funktion nach Bedarf aktivieren bzw. deaktivieren. Informationen hierzu finden Sie unter *Dialogfeld „Einstellungen“* auf Seite 86.

In diesem Fall werden Vor- und Nach-Process-Skripts vor und nach jedem Datensatz ausgeführt. Andernfalls erfolgt die Ausführung vor und nach jedem Process.

Bei neuen Projekten, die in LiveCycle Production Print ES2 erstellt werden, ist diese Funktionalität standardmäßig aktiviert.

In aktualisierten Projekten aus Versionen vor LiveCycle Production Print ES2 werden die Skripts dagegen standardmäßig Process-bezogen ausgeführt.

### Einschränkungen

Die folgenden Skripts können in Vor- und Nach-Process-Skripts nicht verwendet werden, sofern diese in der Process-Phase auf Datensatzebene ausgeführt werden:

- `CallProc`
- `CallBlock`
- `PreProcLog`
- `EndDocument`
- `SetExtJobId`
- `GetJobResourceIndex`
- `AddSortDef`
- `AddSortKeys`
- `ODBCConnect`
- `ODBCDisconnect`
- `SAPCreateFunction`
- `SAPInvokeFunction2`
- `SetDestPath`

# Schreiben in die StreamServe-Protokolldatei

Sie können Protokollmeldungen aus Skripts in einer XDP-Vorlage direkt in das StreamServer-Protokoll schreiben.

### Syntax

`xfa.log.message(`*`Param1`*`, str [, `*`severity`*`])`

| | |
|---|---|
| `Param1` | Dieser Parameter wird von StreamServer grundsätzlich ignoriert. |
| `str` | Meldungszeichenfolge, die in die Protokolldatei geschrieben wird.<br>Der Meldung wird Folgendes vorangestellt:<br>"`XFAOUT: Message from XFA template script: `" |
| `Severity` | Optional. Gibt die Dringlichkeit bzw. den Schweregrad der Meldung an.<br>`i` – Information<br>`t` – Information<br>`w` – Warnung<br>`f` – Fehler |

### Beschreibung

Mithilfe dieser Skriptfunktion können Sie Meldungen aus Skripts in einer XDP-Vorlage direkt in das StreamServer-Protokoll schreiben.

### Beispiel

*Beispiel 13* *Protokollbeispiel*

```
xfa.log.message(1, "text to write in log file", "i");
```

Dabei wird folgende Meldung in das StreamServe-Protokoll geschrieben:

```
0902 175411 (0096) 3 XFAOUT: Message from XFA template script: text
to write in log file
```

# XFA-Unterstützung

Der LiveCycle Production Print XFA-Prozessor implementiert einen Teil der XFA-Spezifikation. Nähere Informationen dazu finden Sie im nachstehenden Abschnitt.

Funktionen, die mit dem Erstellen interaktiver Dokumentausgaben in Verbindung stehen, werden in der Regel nicht unterstützt. Eine Ausnahme bilden die Zeicheneigenschaften solcher Objekte, sofern diese in Druckformaten statisch dargestellt werden.

Das Gleiche gilt für die Skriptimplementierung in FormCalc- und ECMA-Skripts.

**Hinweis:** LiveCycle Production Print ES2 wurde für die Hochleistungsverarbeitung von XFA-Vorlagen entwickelt. Die Funktionsweise des XFA-Prozessors weicht dabei geringfügig von der der entsprechenden XFA-Prozessoren in LiveCycle Output ES2 und LiveCycle Forms ES2 ab. Aus Leistungsgründen gibt es Unterschiede hinsichtlich der Verwendungsmöglichkeiten von Skriptkontexten, wie nachstehend beschrieben.

Informationen zu bekannten Einschränkungen finden Sie in den *Adobe LiveCycle Production Print ES2-Versionshinweisen*.

### In diesem Kapitel behandelte Themen


- *XFA-Elemente* auf Seite 110
- *XFA-Skripts* auf Seite 115

# XFA-Elemente

**In diesem Abschnitt behandelte Themen**


- *Vollständig unterstützte XFA-Elemente* auf Seite 110
- *XFA-Elemente mit nicht unterstützten Attributen* auf Seite 111
- *XFA-Elemente für das Hinzufügen von Daten zu Tagged-PDF-Dateien* auf Seite 113
- *Nicht unterstützte XFA-Elemente* auf Seite 114


## Vollständig unterstützte XFA-Elemente

Die folgenden XFA-Elemente werden vollständig unterstützt. Informationen zu den aktuellen XFA-Spezifikationen finden Sie unter Adobe XML Forms Architecture (XFA).

- `<arc>`
- `<area>`
- `<bind>`
- `<boolean>`
- `<checkButton>`
- `<color>`
- `<comb>`
- `<contentArea>`
- `<corner>`
- `<date>`
- `<dateTime>`
- `<defaultUi>`
- `<draw>`
- `<edge>`
- `<extras>`
- `<fill>`
- `<float>`
- `<font>`
- `<format>`
- `<hyphenation>`
- `<image>`
- `<integer>`
- `<line>`
- `<linear>`
- `<margin>`
- `<occur>`
- `<overflow>`
- `<pattern>`
- `<picture>`
- `<proto>`
- `<radial>`
- `<rectangle>`
- `<solid>`
- `<stipple>`
- `<subformSet>`
- `<time>`
- `<ui>`
- `<variables>`
- `<validate>`

# XFA-Elemente mit nicht unterstützten Attributen

Die folgenden XFA-Elemente werden teilweise unterstützt. Ausführliche Informationen zu den XFA-Elementen finden Sie im Abschnitt *Template Reference* des Dokuments XML Forms Architecture (XFA) (derzeit nur in englischer Sprache verfügbar).

| Element | Nicht unterstützte Attribute |
|---|---|
| <barcode> | • charEncoding<br>• dataPrep<br>• printCheckDigit<br>• rowColumnRatio<br>• upsMode |
| <border> | • break |
| <break> | • bookendLeader<br>• bookendTrailer<br>• after<br>• before |
| <breakAfter> | • leader<br>• trailer<br>• targetType |
| <breakBefore> | • leader<br>• trailer<br>• targetType |
| <button> | • highlight |
| <calculate> | • override |
| <caption> | • placement |
| <choiceList> | • commitOn<br>• open<br>• textEntry |
| <dateTimeEdit> | • hScrollPolicy<br>• picker |
| <decimal> | • leadDigits |
| <event> | • activity |
| <exData> | • maxLength<br>• rid<br>• transferEncoding |
| <exclGroup> | • access<br>• accessKey |

| Element | Nicht unterstützte Attribute |
|---|---|
| <field> | • access<br>• accessKey |
| <imageEdit> | • data |
| <items> | • ref |
| <keep> | • intact<br>• next<br>• previous |
| <medium> | • stock<br>• trayIn<br>• trayOut |
| <numericEdit> | • hScrollPolicy |
| <pageArea> | • blankOrNotBlank<br>• initialNumber<br>• numbered<br>• oddOrEven<br>• pagePosition |
| <pageSet> | • relation |
| <para> | • preserve |
| <passwordEdit> | • hScrollPolicy |
| <script> | • runAt |
| <setProperty> | • connection |
| <subform> | • access<br>• allowMacro<br>• layout<br>• restoreState |
| <template> | • baseProfile |
| <text> | • rid |
| <textEdit> | • allowRichText<br>• hScrollPolicy<br>• vScrollPolicy |
| <value> | • override |

# XFA-Elemente für das Hinzufügen von Daten zu Tagged-PDF-Dateien

Die folgenden XFA-Elemente dienen ausschließlich dem Hinzufügen von Daten zu Dateien im Tagged-PDF-Format. Ausführliche Informationen finden Sie im Abschnitt *Template Reference* des Dokuments XML Forms Architecture (XFA) (derzeit nur in englischer Sprache verfügbar).

- `<assist>`
- `<desc>`
- `<speak>`
- `<toolTip>`
- `<traversal>`
- `<traverse>`

# Nicht unterstützte XFA-Elemente

Die folgenden XFA-Elemente werden nicht unterstützt.

- <appearanceFilter>
- <certificates>
- <digestMethod>
- <encoding>
- <encrypt>
- <execute>
- <handler>
- <keyUsage>
- <manifest>
- <message>
- <oids>
- <reasons>
- <signData>
- <signing>
- <subjectDNs>
- <timeStamp>
- <bindItems>
- <certificate>
- <connect>
- <digestMethods>
- <encodings>
- <exObject>
- <filter>
- <issuers>
- <lockDocument>
- <mdp>
- <oid>
- <reason>
- <ref>
- <signature>
- <subjectDN>
- <submit>

# XFA-Skripts

**In diesem Abschnitt behandelte Themen**


- *Unterstützte Skriptfunktionen* auf Seite 115
- *Unterstützte Skripteigenschaften* auf Seite 118
- *Unterstützte Skriptobjektmodelle* auf Seite 125
- *Unterstützte Events* auf Seite 125
- *Überlegungen zur Verwendung von Skripts* auf Seite 125


## Unterstützte Skriptfunktionen

Die Skriptfunktionen in der nachstehenden Tabelle werden vollständig oder eingeschränkt unterstützt. Nicht in der Tabelle enthaltene Funktionen sind nicht implementiert.

- Für vollständig unterstützte Funktionen gelten die im Handbuch *LiveCycle Scripting Reference Guide* angegebenen Beschreibungen.
- Bei den teilweise unterstützten Funktionen sind die in der nachstehenden Tabelle aufgeführten Ausnahmen zu beachten.

| Funktion | Kommentare |
|---|---|
| `absPage` | Vollständig unterstützt |
| `absPageCount` | Vollständig unterstützt |
| `absPageSpan` | Vollständig unterstützt |
| `addInstance` | • Der Boolesche Parameter `param` wird nicht unterstützt.<br>• Die neue Instanz wird nicht mit dem Daten-DOM zusammengeführt.<br>• Diese Funktion schlägt fehl, wenn der Wert des Attributs `maxOccurs` eines Unterformulars im gegebenen Kontext durch Hinzufügen des Unterformulars überschritten wird. |
| `addItem` | Vollständig unterstützt |
| `assignNode` | • Diese Funktion ist nur für Knoten im Formular-DOM oder im Layout-DOM verfügbar.<br>• Der Parameter `param3` wird nicht unterstützt.<br>• Das Erstellen neuer Knoten ist nicht möglich.<br>• Diese Funktion versucht stets, den Wert einem vorhandenen Knoten zuzuweisen, andernfalls schlägt sie fehl. |
| `boundItem` | Vollständig unterstützt |

| Funktion | Kommentare |
|---|---|
| `clearItems` | Vollständig unterstützt |
| `clone` | Diese Funktion ist nur für Knoten im Formular-DOM oder im Layout-DOM verfügbar. |
| `deleteItem` | Vollständig unterstützt |
| `execCalculate` | Diese Funktion wird nur für `exclGroup`, `field`, `form` und `subform` unterstützt. |
| `execEvent` | Vollständig unterstützt |
| `execInitialize` | Diese Funktion wird nur für `exclGroup`, `field`, `form` und `subform` unterstützt. |
| `getAttribute` | Vollständig unterstützt |
| `getDisplayItem` | Vollständig unterstützt |
| `getElement` | Vollständig unterstützt |
| `getItemState` | Vollständig unterstützt |
| `getSaveItem` | Vollständig unterstützt |
| `h` | • Diese Funktion wird nur für `field`, `draw`, `subform` und `contentArea` unterstützt.<br>• Der Parameter `param3` muss 0 sein, andernfalls gibt diese Funktion 0 zurück. |
| `isCompatibleNS` | • Für Objekte im Vorlagen-DOM, im Formular-DOM und im Layout-DOM gibt diese Funktion für die folgenden Namespaces `true` zurück:<br>`http://www.xfa.org/schema/xfa-template/`<br>`http://www.xfa.org/schema/xfa-template/2.1/`<br>`http://www.xfa.org/schema/xfa-template/2.4/`<br>`http://www.xfa.org/schema/xfa-template/2.5/`<br>`http://www.xfa.org/schema/xfa-template/2.6/`<br>• Für Objekte im Daten-DOM gibt diese Funktion für die folgenden Namespace `true` zurück:<br>`http://www.xfa.org/schema/xfa-data/`<br>• Für alle anderen Namespace-Objekt-Modellkombinationen wird `false` zurückgegeben. |
| `isPropertySpecified` | Vollständig unterstützt |
| `item` | Vollständig unterstützt |
| `message` | Vollständig unterstützt |

| Funktion | Kommentare |
|---|---|
| `messageBox` | • Nur in Java-Skript implementiert, es wird jedoch kein Nachrichtenfeld angezeigt.<br>• Methode gibt immer 2 (Abbrechen) zurück. |
| `namedItem` | Vollständig unterstützt |
| `page` | Vollständig unterstützt |
| `pageContent` | • Keine Unterstützung für das Abrufen des gesamten Seiteninhalts (z. B. Angeben eines leeren Werts für `param2`).<br>• Keine Unterstützung für `param3`.<br>• Diese Funktion durchsucht stets die gesamte Seite einschließlich `pageArea` und `contentAreas`. |
| `pageCount` | Vollständig unterstützt |
| `pageSpan` | Vollständig unterstützt |
| `relayout` | • Eingeschränkte Funktionalität. Nach Abschluss des aktuellen Layouts wird ein zweiter Layoutdurchlauf ausgeführt. Beim zweiten Durchlauf werden alle Seiten mit allen Objekten vollständig neu erstellt.<br>• Skripts auf Formularknoten werden nach dieser Layoutneuerstellung nicht erneut ausgeführt. Lediglich Skripts auf Überlaufkopf- und -fußzeilen sowie auf Boilerplate-Objekten werden ausgeführt. |
| `remerge` | Durch diese Funktion wird das Formular-DOM in einem einmaligen Durchgang vollständig neu erstellt. Wenn `remerge` in einem zweiten Durchgang erneut aufgerufen wird, hat dies keinerlei Auswirkungen. |
| `removeInstance` | Vollständig unterstützt |
| `resolveNode` | Vollständig unterstützt |
| `resolveNodes` | Vollständig unterstützt |
| `saveXML` | Diese Funktion ist implementiert, es kann jedoch keine Garantie für die Gültigkeit der zurückgegebenen XML gegeben werden. |
| `selectedMember` | Vollständig unterstützt |
| `setAttribute` | Vollständig unterstützt |
| `setElement` | Diese Funktion ist implementiert, funktioniert jedoch nicht zuverlässig. Ein Ersetzen von Elementen wird nicht immer in die Ausgabe integriert. |
| `setItems` | Vollständig unterstützt |

| Funktion | Kommentare |
|---|---|
| `setItemState` | Vollständig unterstützt |
| `sheet` | Diese Funktion gibt immer den gleichen Wert wie `page` zurück. |
| `sheetCount` | Diese Funktion gibt immer den gleichen Wert wie `page` zurück. |
| `w` | • Diese Funktion wird nur für `field`, `draw`, `subform`, `contentArea` und `pageArea` unterstützt.<br>• Der Parameter `param3` muss 0 sein, andernfalls gibt diese Funktion 0 zurück. |
| `x` | Der Parameter `param3` muss 0 sein, andernfalls gibt diese Funktion 0 zurück. |
| `y` | Der Parameter `param3` muss 0 sein, andernfalls gibt diese Funktion 0 zurück. |

## Unterstützte Skripteigenschaften

Die Skripteigenschaften in der nachfolgenden Tabelle werden vollständig oder eingeschränkt unterstützt. Nicht in der Tabelle enthaltene Eigenschaften sind nicht implementiert.

- Für vollständig unterstützte Eigenschaften gelten die im Handbuch *LiveCycle Scripting Reference Guide* angegebenen Beschreibungen.
- Bei den teilweise unterstützten Eigenschaften sind die in der nachstehenden Tabelle aufgeführten Ausnahmen zu beachten.

| Eigenschaft | Kommentare |
|---|---|
| `#text` | Vollständig unterstützt |
| `{default}` | Vollständig unterstützt |
| `activity` | Diese Eigenschaft ist schreibgeschützt und kann nicht per Skriptzugriff geändert werden. |
| `after` | Vollständig unterstützt |
| `afterTarget` | Wenn Sie diese Eigenschaft festlegen, können nur Namen von in der aktuellen Vorlage vorhandenen Unterformularen verwendet werden. |
| `all` | Vollständig unterstützt |
| `allowNeutral` | Vollständig unterstützt |
| `allowRichText` | Vollständig unterstützt |

| Eigenschaft | Kommentare |
|---|---|
| anchorType | Vollständig unterstützt |
| aspect | Vollständig unterstützt |
| baselineShift | Vollständig unterstützt |
| before | Vollständig unterstützt |
| beforeTarget | Wenn Sie diese Eigenschaft festlegen, können nur Namen von in der aktuellen Vorlage vorhandenen Unterformularen verwendet werden. |
| borderColor | Vollständig unterstützt |
| borderWidth | Vollständig unterstützt |
| bottomInset | Vollständig unterstützt |
| break | Vollständig unterstützt |
| cap | Vollständig unterstützt |
| charEncoding | Vollständig unterstützt |
| checksum | Vollständig unterstützt |
| circular | Vollständig unterstützt |
| classAll | Vollständig unterstützt |
| classIndex | Vollständig unterstützt |
| className | Vollständig unterstützt |
| colSpan | Vollständig unterstützt |
| columnWidths | Vollständig unterstützt |
| contentType | Diese Eigenschaft wird für die folgenden Objekte unterstützt:<br>• exData (Lesen/Schreiben)<br>• image (Lesen/Schreiben)<br>• script (schreibgeschützt) |
| count | Vollständig unterstützt |
| cSpace | Vollständig unterstützt |
| data | Vollständig unterstützt |
| dataColumnCount | Vollständig unterstützt |
| dataLength | Vollständig unterstützt |
| dataRowCount | Vollständig unterstützt |

| Eigenschaft | Kommentare |
|---|---|
| editValue | Diese Eigenschaft ruft rawValue ab oder legt diesen Wert fest. Die Unterscheidung erfolgt zwischen rawValue und editValue. |
| endChar | Vollständig unterstützt |
| errorCorrectionLevel | Vollständig unterstützt |
| excludeAllCaps | Da die Silbentrennung in LiveCycle und StreamServe unterschiedlich implementiert ist, kann nicht davon ausgegangen werden, dass Text mit Silbentrennung identisch dargestellt wird. |
| excludeInitialCap | Da die Silbentrennung in LiveCycle und StreamServe unterschiedlich implementiert ist, kann nicht davon ausgegangen werden, dass Text mit Silbentrennung identisch dargestellt wird. |
| fillColor | Vollständig unterstützt |
| fontColor | Vollständig unterstützt |
| fontHorizontalScale | Vollständig unterstützt |
| fontVerticalScale | Vollständig unterstützt |
| formattedValue | Mit dem Festlegen dieser Eigenschaft wird auch die Eigenschaft rawValue festgelegt. Bei Abrufen dieser Eigenschaft wird rawValue vor der Rückgabe an den Aufrufer formatiert. |
| fracDigits | Vollständig unterstützt |
| h | Vollständig unterstützt |
| hAlign | Vollständig unterstützt |
| hand | Vollständig unterstützt |
| highlight | Vollständig unterstützt |
| href | Nur in image-Objekten verwendet. Kann auch in exData festgelegt und gelesen werden, hat in diesem Objekt jedoch keinerlei Bedeutung. |
| id | Schreibgeschützt |
| index | Schreibgeschützt |
| initial | Vollständig unterstützt |
| instanceIndex | Schreibgeschützt |
| intact | Vollständig unterstützt |
| inverted | Vollständig unterstützt |
| isContainer | Vollständig unterstützt |

| Eigenschaft | Kommentare |
|---|---|
| isNull | Vollständig unterstützt |
| join | Vollständig unterstützt |
| kerningMode | Vollständig unterstützt |
| ladderCount | Vollständig unterstützt |
| layout | Vollständig unterstützt |
| leadDigits | Vollständig unterstützt |
| leader | Unterstützt für breakAfter, breakBefore und overflow. |
| leftInset | Vollständig unterstützt |
| length | Vollständig unterstützt |
| letterSpacing | Vollständig unterstützt |
| lineHeight | Vollständig unterstützt |
| lineThrough | Vollständig unterstützt |
| lineThroughPeriod | Vollständig unterstützt |
| listen | Schreibgeschützt |
| locale | Vollständig unterstützt |
| long | Vollständig unterstützt |
| marginLeft | Vollständig unterstützt |
| marginRight | Vollständig unterstützt |
| mark | Vollständig unterstützt |
| bind | Schreibgeschützt |
| max | Vollständig unterstützt |
| maxChars | Vollständig unterstützt |
| maxH | Vollständig unterstützt |
| maxW | Vollständig unterstützt |
| min | Vollständig unterstützt |
| minH | Vollständig unterstützt |
| minW | Vollständig unterstützt |
| moduleHeight | Vollständig unterstützt |
| moduleWidth | Vollständig unterstützt |

| Eigenschaft | Kommentare |
|---|---|
| multiLine | Vollständig unterstützt |
| name | Vollständig unterstützt |
| next | Die Option pageArea wird nicht unterstützt. |
| nodes | Vollständig unterstützt |
| ns | Schreibgeschützt |
| numberOfCells | Vollständig unterstützt |
| numPages | Schreibgeschützt |
| orientation | Vollständig unterstützt |
| overflowLeader | Schreibgeschützt |
| overflowTarget | Schreibgeschützt |
| overflowTrailer | Schreibgeschützt |
| override | Schreibgeschützt. Kann festgelegt werden, hat jedoch nach dem Laden der Vorlage keinerlei Auswirkungen. |
| parent | Schreibgeschützt |
| parentSubform | Schreibgeschützt |
| passwordChar | Vollständig unterstützt |
| placement | Vollständig unterstützt |
| posture | Vollständig unterstützt |
| presence | Vollständig unterstützt |
| preserve | Vollständig unterstützt |
| previous | Die Option pageArea wird nicht unterstützt. |
| printCheckDigit | Vollständig unterstützt |
| pushCharacterCount | Da die Silbentrennung in LiveCycle und StreamServe unterschiedlich implementiert ist, kann nicht davon ausgegangen werden, dass Text mit Silbentrennung identisch dargestellt wird. |
| radius | Vollständig unterstützt |
| radixOffset | Vollständig unterstützt |
| rate | Vollständig unterstützt |

| Eigenschaft | Kommentare |
|---|---|
| rawValue | formattedValue und editValue entsprechen rawValue in Production Print. Das Festlegen eines dieser Werte führt dazu, dass alle auf den gleichen Wert festgelegt werden. formatted-Value wird jedoch bei der Ausgabe erneut formatiert. Dies ist ein bekanntes Problem. |
| ref | Schreibgeschützt |
| relation | Schreibgeschützt |
| relevant | Vollständig unterstützt |
| remainCharacterCount | Da die Silbentrennung in LiveCycle und StreamServe unterschiedlich implementiert ist, kann nicht davon ausgegangen werden, dass Text mit Silbentrennung identisch dargestellt wird. |
| reserve | Vollständig unterstützt |
| rightInset | Vollständig unterstützt |
| rotate | Vollständig unterstützt |
| runAt | Wie schreibgeschützt behandeln. Kann festgelegt werden, hat jedoch keinerlei Auswirkungen. |
| save | Vollständig unterstützt |
| scope | Schreibgeschützt |
| selectedIndex | Vollständig unterstützt |
| shape | Vollständig unterstützt |
| short | Vollständig unterstützt |
| size | Vollständig unterstützt |
| slope | Vollständig unterstützt |
| somExpression | Schreibgeschützt |
| spaceAbove | Vollständig unterstützt |
| spaceBelow | Vollständig unterstützt |
| startAngle | Vollständig unterstützt |
| startChar | Vollständig unterstützt |
| startNew | Vollständig unterstützt |
| stroke | Vollständig unterstützt |
| sweepAngle | Vollständig unterstützt |
| tabDefault | Vollständig unterstützt |

| Eigenschaft | Kommentare |
|---|---|
| target | Unterstützt für breakAfter und breakBefore. Wenn Sie diese Eigenschaft festlegen, können nur Namen von in der aktuellen Vorlage vorhandenen Unterformularen verwendet werden. |
| targetType | Keine Unterstützung für die Optionen pageEven und pageOdd. |
| textIndent | Vollständig unterstützt |
| textLocation | Vollständig unterstützt |
| thickness | Vollständig unterstützt |
| this | Vollständig unterstützt |
| topInset | Vollständig unterstützt |
| trailer | Unterstützt für breakAfter und breakBefore. Wenn Sie diese Eigenschaft festlegen, können nur Namen von in der aktuellen Vorlage vorhandenen Unterformularen verwendet werden. |
| transferEncoding | Nur für image unterstützt. |
| truncate | Vollständig unterstützt |
| type | Unterstützt für barcode, linear, pattern und radial. |
| typeface | Unterstützt, die entsprechende Schriftart muss jedoch in das StreamServe-Projekt eingeschlossen werden. |
| underline | Vollständig unterstützt |
| underlinePeriod | Vollständig unterstützt |
| use | Schreibgeschützt |
| usehref | Schreibgeschützt |
| vAlign | Vollständig unterstützt |
| value | Gibt für Objekte, die rawValue, value und rawValue unterstützen, das gleiche Ergebnis zurück und legt den gleichen Wert fest. |
| w | Vollständig unterstützt |
| weight | Vollständig unterstützt |
| wideNarrowRatio | Vollständig unterstützt |
| wordCharacterCount | Da die Silbentrennung in LiveCycle und StreamServe unterschiedlich implementiert ist, kann nicht davon ausgegangen werden, dass Text mit Silbentrennung identisch dargestellt wird. |
| x | Vollständig unterstützt |
| y | Vollständig unterstützt |

# Unterstützte Skriptobjektmodelle

Die Skriptobjektmodelle in der nachfolgenden Liste werden vollständig oder eingeschränkt unterstützt. Informationen zur eingeschränkten Unterstützung finden Sie unter *Unterstützte Skriptfunktionen* auf Seite 115 und *Unterstützte Skripteigenschaften* auf Seite 118.

Nicht in der Liste aufgeführte Objektmodelle werden nicht unterstützt.

- `Datenmodell`
- `Formularmodell`
- `Hostmodell`
- `Layoutmodell`
- `Protokollmodell`
- `XFA-Modell`

# Unterstützte Events

Die Formular-Events in der nachstehenden Liste werden vollständig unterstützt. Für vollständig unterstützte Events gelten die im Handbuch *LiveCycle Designer ES2 Scripting Basics* angegebenen Beschreibungen.

Nicht in der Tabelle enthaltene Events sind nicht implementiert.

- `calculate`
- `form:ready`
- `indexChange`
- `initialize`
- `layout:ready`
- `prePrint`
- `validate`

# Überlegungen zur Verwendung von Skripts

### Skripts im Layout-DOM

Das Layout-DOM wird zuerst vollständig erstellt und über Seiten, Inhaltsbereiche usw. verteilt. Anschließend werden alle Layout-ready- und Preprint-Skripts in der Reihenfolge ausgeführt, die in der XFA 3.1-Spezifikation angegeben ist. Wenn ein Layout-ready- oder Preprint-Skript das Layout-DOM so ändert, dass die Objekte auf eine andere Seite wechseln, wird das Layout-DOM vollständig neu erstellt.

Beim Neuerstellen des Layout-DOM werden folgende Elemente gelöscht:

- sämtliche dynamisch erstellten Layoutobjekte (wie Überlauf-Leader und -Trailer)
- sämtliche Objekte auf Masterseiten

Alle anderen Layoutobjekte (wie Felder aus dem Formular-DOM) und deren Werte bleiben erhalten.

Um die richtigen Werte beizubehalten, werden nach der Neuerstellung des Layout-DOM keine Layout-Skripts für Formularobjekte ausgeführt. Zu diesem Zeitpunkt werden nur Skripts ausgeführt, die den dynamisch erstellten Layoutobjekten (Überlauf-Leader, Objekte auf Masterseiten usw.) angehängt sind.

Die Layout-DOM-Skripts werden mit den folgenden wenigen Ausnahmen für jedes Objekt nur einmal ausgeführt:

- Die Layout-ready- und Preprint-Skripts für das Stammunterformular werden sowohl bei der ersten als auch ggf. bei der zweiten Erstellung und Anpassung des Layout-DOM ausgeführt.
- Initialisierungs- und Berechnungsskripts für Objekte auf Masterseiten können zweimal ausgeführt werden.

Folglich müssen Sie bei der Verwendung bzw. Initialisierung von globalen Variablen oder beim Zugreifen auf Felder und andere Werte in dynamisch erstellten Unterformularen in Objekten im Formular-DOM bedacht vorgehen:

- Jegliche Initialisierung von globalen Variablen und jegliche extern (aus anderen Teilen des DOM) an dynamisch erstellten Layoutobjekten vorgenommenen Änderungen müssen im Stammunterformular oder in Skriptfunktionen vorgenommen werden, die aus dem Stammunterformular aufgerufen werden.
- Das Aktualisieren von Werten in Formular-DOM-Objekten, beispielsweise aus einer Überlaufkopfzeile, kann zu unerwarteten Ergebnissen führen.

  Beispiel: `subform.field.x = subform.field.x + 100;` in einem Überlauf-Leader führt dazu, dass `subform.field.x` bei der ersten Erstellung des Layout-DOM um 100 erhöht wird. Wenn das Layout-DOM noch einmal neu erstellt wird, werden `x` weitere 100 hinzugefügt.

### Inline-deklarierte globale Variablen in Java-Skripts

- Inline-deklarierte globale Variablen in Java-Skripts im Formular-DOM, z. B. `initialize` oder `form ready`, können mit den initialisierten Werten nicht im Layout-DOM verwendet werden. Hierzu besteht eine Ausnahme: in dedizierten Skriptobjekten deklarierte globale Variablen. Diese Variablen können in Formular-DOM-Skripts initialisiert und in Layout-DOM-Skripts verwendet werden.
- Inline-deklarierte globale Variablen in Java-Skripts können nicht in FormCalc-Skripts und SOM-Ausdrücken verwendet werden (beispielsweise zur Übertragung von Werten an StreamServe-Variablen). Um diese Variablen verwenden zu können, müssen sie als Formularvariablen deklariert sein (d. h., sie müssen in der Hierarchiestruktur in LiveCycle Designer ES2 sichtbar sein).

### Reihenfolge der Ausführung von Skripts

Die Ausführungsreihenfolge für Skripts in Bezug auf Objekte eines Event-Typs in XFA ist nicht definiert. Schreiben Sie daher keine Skripts, für die eine bestimmte Ausführungsreihenfolge wichtig ist.

# Glossar

Einige in diesem Dokument verwendete Begriffe und Konzepte:

**Abhängigkeiten**
Fragmente und Bilder, auf die aus einer XDP-Datei, der XDP-Hauptdatei, verwiesen wird.

**Bereitstellen eines Projekts**
Beim Bereitstellen eines Projekts werden die Konfigurationsdateien mithilfe der Exportdatei entpackt und im Arbeitsverzeichnis der StreamServer-Anwendung gespeichert.

**Control Center**
Mit StreamServe Control Center werden StreamServe-Anwendungen verwaltet und administriert.

**Design Center**
Das StreamServe-Konfigurationstool für Ihre StreamServe-Projekte.

**Document**
Bei einem „Document“ (englische Schreibweise) handelt es sich um eine Gruppierung mehrerer Dokumente nach Kundennummer, Lieferadresse usw. Der Umfang des Documents wird mit einem Document-Trigger festgelegt.

**Event**
Ein Event steuert, wie Felder erkannt und aus den Eingabedaten extrahiert und wie die extrahierten Felder strukturiert und bezeichnet werden.

**Exportieren eines Projekts**
Beim Exportieren eines Projekts erzeugen Sie eine Paketdatei (`*.export`), die sämtliche für die Ausführung des Projekts erforderlichen Konfigurationsdateien enthält. Der Export erfolgt aus Design Center.

**LCA**
LiveCycle-Archiv.

**LiveCycle Designer-Process-Tool**
Die von StreamServe entwickelte Anwendung, mit der die Adobe LiveCycle Designer ES2-Anwendung in StreamServe Design Center integriert wird.

**LiveCycle ES2**
Adobe LiveCycle ES2 (Enterprise Suite)

**Message**
Über eine Message wird die Umwandlung vom Eingabe- in das Ausgabeformat verwaltet. Ein Projekt enthält in der Regel mehrere Messages, wobei jede Message einem bestimmten Dokumenttyp entspricht. So gibt es beispielsweise eine Message für Rechnungen, eine weitere für Aufträge usw.

Eine Message enthält Events und Processes.

**PageIN** Das Event-Tool für die Verarbeitung seitenformatierter Eingabedaten. Informationen zu weiteren Event-Tools finden Sie in der Dokumentation zu *StreamServe Design Center*.

**PageOUT** Das Process-Tool, das seitenorientierte Ausgabedaten erstellt. Informationen zu weiteren Process-Tools finden Sie in der Dokumentation zu *StreamServe Design Center*.

**Platform** In der Platform konfigurieren Sie die Umgebungseinstellungen für ein Projekt. So geben Sie beispielsweise in der Platform die Verbindungen zu den Quellanwendungen und Ausgabegeräten (Drucker, Faxgeräte usw.) an und wie Eingabedaten empfangen und Ausgabedaten übermittelt werden.

**Process** Ein Process steuert, welche Felder abgerufen werden und wie die Felder in der StreamServer-Ausgabe strukturiert werden.

**Process-Tool** Process-Tool für LiveCycle Designer ES2. Das von StreamServe entwickelte Tool, mit dem die Adobe LiveCycle Designer ES2-Anwendung in StreamServe Design Center integriert wird.

**Projekt** Über ein StreamServe-Projekt wird definiert, wie eine spezifische EDP-Lösung Daten erfassen, umwandeln und übermitteln soll.

**Ressource** Eine Ressource ist eine Datei mit einer eingebetteten Quelldatei. Um beispielsweise die Bilddatei `logo.gif` im Projekt verwenden zu können, muss diese Datei in einer Ressource eingebettet sein.

Alle externen Dateien, außer Abhängigkeiten, auf die Sie bei der Konfiguration eines Projekts verweisen, müssen in Ressourcen umgewandelt werden.

**Ressourcenset** Ein Satz von Verknüpfungen zu Ressourcendateien.

**Runtime** In der Runtime-Konfiguration verbinden Sie Messages mit der Platform. Ein Projekt enthält meist mehrere Runtime-Konfigurationen, z. B. eine Runtime-Konfiguration je Message.

**StreamIN** Das Event-Tool für die Verarbeitung feld- oder datensatzbasierter Eingabedaten.

| | |
|---|---|
| **StreamServer** | Die für die Erfassung und Umwandlung der Eingabedaten sowie die Übermittlung der Ausgabedokumente verantwortliche Komponente. StreamServer kann mehrere StreamServer-Anwendungen ausführen, wobei jede StreamServer-Anwendung für ein bestimmtes StreamServe-Projekt zuständig ist. |
| **XDP** | XML-Datenpaket. |
| **XFA** | XML-Formulararchitektur. |
| **XFA-Prozessor** | Die von StreamServe entwickelte Anwendung, die es der StreamServer-Laufzeitumgebung ermöglicht, die Ausgabe von XML-Datendateien und Adobe LiveCycle Designer ES2-Formularvorlagen in verschiedenen Formaten zu generieren. |
| **XMLIN** | Das Event-Tool für die Verarbeitung XML-formatierter Eingabedaten. |

# Glossar